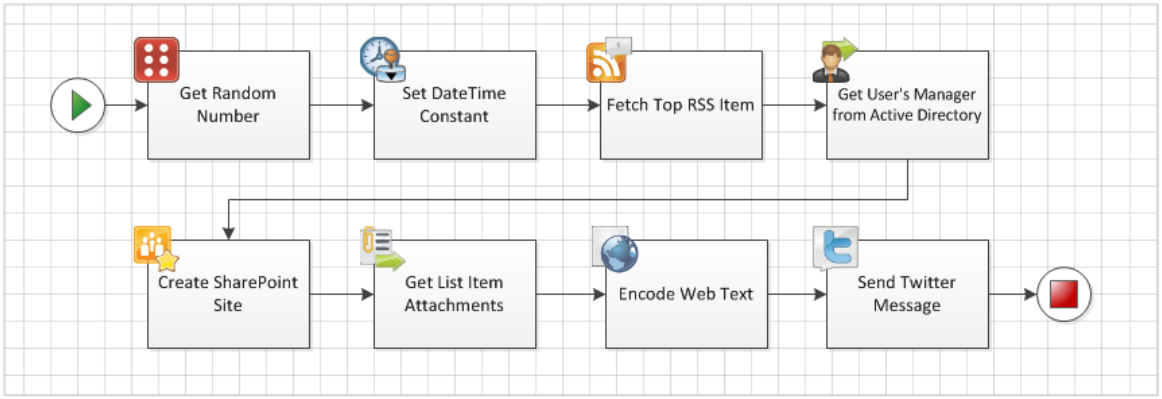
one letter doesn't matter
HarePoint
Web-parts and tools for SharePoint

HarePoint Workflow Extensions Manual

For SharePoint Server 2013, SharePoint Server 2010, SharePoint Foundation 2013 and SharePoint Foundation 2010.



Product version 2.3

7/14/2015

(This Page Intentionally Left Blank)

Table of Contents

Table of Contents 3

I. Introduction 14

II. Deployment 15

1. System requirements 15

2. Licensing and trial limitations 16

3. Installing HarePoint Workflow Extensions 17

4. Installing Client Applications 20

5. How to repair, upgrade or remove product 22

III. Product Overview 25

1. Getting started 25

2. Creating workflows in SharePoint Designer 26

3. Designing workflows in Microsoft Visio 27

4. Creating workflows in Visual Studio 28

5. Security issues 29

6. How to disable certain HarePoint activities 30

IV. HarePoint Activities and Conditions 32

1. Active Directory activities 32

1.1 Add Member to Active Directory Group 32

1.2 Change Active Directory Account Settings 32

1.3 Change Local User Account Settings on Server 33

1.4 Create Group in Active Directory 34

1.5 Create Local User Account on Server 34

1.6 Create Mail Contact in Active Directory 35

1.7 Create User in Active Directory 36

1.8 Delete Group from Active Directory 36

1.9 Delete Local User Account from Server 37

1.10 Delete User from Active Directory 37

1.11 Find User in Active Directory 38

1.12 Find User in Active Directory by Query 38

1.13 Get Active Directory Groups Where User is Member 39
1.14 Get E-Mails of Active Directory Group Members 39
1.15 Get User's Attribute from Active Directory 40
1.16 Get User's Manager from Active Directory 40
1.17 Get User's Multivalued Attribute from Active Directory 41
1.18 Is User a Member of an Active Directory Group (HarePoint) 41
1.19 Move User into Organizational Unit 42
1.20 Remove Member from Active Directory Group 42
1.21 Set Active Directory Authentication 43
1.22 Set Local User's Password on Server 44
1.23 Set User's Attribute in Active Directory 44
1.24 Set User's Multivalued Attribute in Active Directory 45
1.25 Set User's Password in Active Directory 45
1.26 Set User's Photo in Active Directory 46
2. Administration activities 46
2.1 Activate Feature 46
2.2 Add Users to SharePoint Group 46
2.3 Add Users to Site 47
2.4 Change SharePoint Group Owner 47
2.5 Create SharePoint Group 47
2.6 Create SharePoint Site 48
2.7 Create Site Collection 49
2.8 Deactivate Feature 49
2.9 Delete SharePoint Group 50
2.10 Delete SharePoint Site 50
2.11 Delete Site Collection 50
2.12 Get Multiple Site Users Property 51
2.13 Get Site User Property 51
2.14 Is Role Assigned to User (HarePoint) 52
2.15 Is User a Member of a SharePoint Group (HarePoint) 53
2.16 Remove User from SharePoint Group 53

2.17 Remove User from Site 53
2.18 Rename Site at Url 54
2.19 Set Master Page 54
2.20 Set SharePoint List Anonymous Access 54
2.21 Set SharePoint Site Anonymous Access 55
2.22 Set SharePoint Site Navigation Options 55
2.23 Set Site User Property 56
2.24 Set User's Photo in SharePoint 56
3. Arrays activities 57
3.1 Add Array Item 57
3.2 Clear Array 57
3.3 Create Array 57
3.4 Get Array Item 58
3.5 Get Array Items Count 58
3.6 Get Index of Array Item 59
3.7 Insert Array Item 60
3.8 Remove Array Item 60
3.9 Remove Array Item at 61
3.10 Remove Duplicates from Array 61
4. Date/Time activities 61
4.1 Converts Date from Local Time to UTC 61
4.2 Converts Date from UTC to Local Time 61
4.3 Converts Date to Text 62
4.4 Evaluate Due Date 63
4.5 Get Date Property 63
4.6 Get Interval Between Dates 65
4.7 Set DateTime Constant 65
5. Development activities 66
5.1 Execute Custom Code 66
5.2 Execute PowerShell Script 68
5.3 Execute SQL Command 69

5.4 Execute SQL Command into List 71
5.5 Query XML 72
5.6 Transform XML 74
5.7 Update XML 76
6. Document Conversion activities 78
6.1 Convert Document 78
6.2 Convert Folder at Url 79
6.3 Convert InfoPath Form and Save to Document Library 80
6.4 Convert Library 81
6.5 Get Conversion Job Status 82
6.6 Wait Conversion Job 82
7. Document Library activities 83
7.1 Add Watermark To PDF Document 83
7.2 Check In Item at Url 84
7.3 Check Out Item at Url 84
7.4 Copy Document to List Item Attachment 84
7.5 Copy or Move Document 85
7.6 Copy or Move Document to Different Farm 86
7.7 Copy or Move Folder 86
7.8 Create Document from File 87
7.9 Create Document from Text 87
7.10 Create Document Set in the Document Library 88
7.11 Create Folder in the Document Library 88
7.12 Create Folder in the Document Library at Url 89
7.13 Create Link to a Document 89
7.14 Delete Folder in the Document Library 89
7.15 Delete Folder in the Document Library at Url 90
7.16 Discard Check Out Item at Url 90
7.17 Discard Publish Item 90
7.18 Download Documents to Folder 90
7.19 Get Document Content 91

7.20 Get Entry From Folder Property Bag 91
7.21 Publish Item 92
7.22 Rename Document at Url 92
7.23 Save Document to File 92
7.24 Set Network File Access 93
7.25 Set PDF Watermark Options 94
7.26 Set Target Site Options 95
7.27 Upload Documents from Folder 95
8. E-Mail activities 96
8.1 Add Custom Headers To Email 96
8.2 Get E-mail Property 97
8.3 Receive All E-Mails to List from IMAP Server 97
8.4 Receive All E-Mails to List from POP3 Server 98
8.5 Receive E-Mail from IMAP Server 100
8.6 Receive E-Mail from POP3 Server 101
8.7 Save E-Mail Attachments to List Item 102
8.8 Send E-Mail with Document as Attachment 102
8.9 Send E-Mail with File Attachments 103
8.10 Send E-Mail with Http File Attachment 104
8.11 Send E-Mail with InfoPath Attachments 105
8.12 Send E-Mail with InfoPath Form 106
8.13 Send E-Mail with List Item Attachments 107
8.14 Send E-Mail with Multiple Attachments 107
8.15 Send E-Mail without Attachments 108
8.16 Send Meeting Request 109
8.17 Send SMS Message 109
8.18 Set SMTP Server Options 110
9. Images activities 111
9.1 Convert Image 111
9.2 Convert Images in Folder 112
9.3 Convert Images in Library 112

9.4 Flip Image 113
9.5 Get EXIF Property from Image 113
9.6 Get Image Size 115
9.7 Get XMP Property 116
9.8 Is Image Compressed (HarePoint) 117
9.9 Is Image Type Supported (HarePoint) 117
9.10 Resize Image 119
9.11 Resize Images in Folder 120
9.12 Resize Images in Library 121
9.13 Rotate Image 122
9.14 Set EXIF Property on Image 122
9.15 Set Transparent Color 122
9.16 Set XMP Property 123
10. Lists activities 124
10.1 Add Content Type to List 124
10.2 Add Existing Column to Content Type 124
10.3 Approve Item at Url 124
10.4 Build String from List 125
10.5 Calculate Aggregation Function 126
10.6 Copy InfoPath Attachments to Document Library 128
10.7 Copy List Item at Url 130
10.8 Copy List Item Attachments to Document Library 130
10.9 Copy List Item Attachments to Document Library (advanced) 131
10.10 Copy List Item Attachments to List Item 131
10.11 Copy List Item to Different Farm 132
10.12 Create Content Type 132
10.13 Create List 133
10.14 Create List Item at Url 133
10.15 Create List Item on Different Farm 135
10.16 Create Site Column 136
10.17 Delete All List Items 137

10.18 Delete Item at Url 138
10.19 Delete Item on Different Farm 139
10.20 Download InfoPath Attachments to Folder 139
10.21 Find List Item at Url 140
10.22 Find List Item by CAML 141
10.23 Find List Item by CAML on Different Farm 141
10.24 Find List Item on Different Farm 142
10.25 Get Field Value at Url 143
10.26 Get List Item Attachment Content 143
10.27 Get List Item Attachments 144
10.28 Get List Item Url 144
10.29 Get List Url 145
10.30 Get Workflow Task Property 146
10.31 If List Exists (HarePoint) 146
10.32 Is Current Item Checked Out (HarePoint) 147
10.33 Is Specified Item Checked Out (HarePoint) 147
10.34 Reject Item at Url 147
10.35 Remove All Attachments 148
10.36 Remove Content Type from List 148
10.37 Remove InfoPath Attachments 148
10.38 Set Content Type Document Template 149
10.39 Set Default Content Type on List 149
10.40 Update List Item at Url 150
10.41 Update List Item on Different Farm 151
10.42 Update Workflow Task at Url 152
10.43 Upload Files to List Item Attachments 153
10.44 Wait for Field Change in Current Item Until Date 153
10.45 Wait for Field Change in Current Item With Timeout 154
11. Network activities 154
11.1 Create Directory on FTP Server 154
11.2 Delete Directory from FTP Server 155

11.3 Delete File from FTP Server 155
11.4 Fetch RSS Feed to Html 155
11.5 Fetch RSS Item 156
11.6 Send HTTP GET Request 156
11.7 Send HTTP OAuth Request 157
11.8 Send HTTP POST Request 159
11.9 Send XMPP Instant Message 160
11.10 Set Encrypted Passwords 160
11.11 Set Exchange Web Service Options 161
11.12 Set FTP Server Options 162
11.13 Set HTTP Request Options 163
11.14 Set OAuth Authentication 164
11.15 Set Web Service Options 165
11.16 Shorten Url 165
11.17 Upload Document to FTP Server 166
11.18 Upload List Item Attachments to FTP Server 166
12. Office activities 167
12.1 Accept Revision Changes in Word Document 167
12.2 Add New Row into Excel Document 168
12.3 Add New Row into Word Document 168
12.4 Find Text in Word Document 169
12.5 Get Cell Value 169
12.6 Get Office Document Property 170
12.7 Read Word Document 171
12.8 Read Word Document To Text 171
12.9 Replace Text in Word Document 172
12.10 Set Cell Value 172
12.11 Set Office Document Property 173
12.12 Update Picture in Word Document 174
12.13 Update Word Document 174
13. Permissions activities 176

13.1 Change Permissions Inheritance at Url 176
13.2 Change Permissions Inheritance for List 176
13.3 Change Permissions Inheritance for List Item 176
13.4 Delete All Permissions at Url 177
13.5 Delete All Permissions from List 177
13.6 Delete All Permissions from List Item 177
13.7 Delete All User Permissions at Url 178
13.8 Delete All User Permissions from List 178
13.9 Delete All User Permissions from List Item 179
13.10 Delete Permissions at Url 179
13.11 Delete Permissions from List 180
13.12 Delete Permissions from List Item 181
13.13 Grant Permissions at Url 182
13.14 Grant Permissions on List 182
13.15 Grant Permissions on List Item 183
14. Printing activities 184
14.1 Insert Watermark into PDF Before Printing 184
14.2 Print Document 186
14.3 Set Printing Options 186
15. Social activities 186
15.1 Create Group in Yammer 186
15.2 Create MailChimp Campaign 187
15.3 Create User in Yammer 188
15.4 Delete MailChimp Campaign 189
15.5 Delete User from Yammer 189
15.6 Get User's Property from Yammer 189
15.7 Invite User to Yammer 190
15.8 Publish Message to Social Media via dlvr.it 190
15.9 Publish Message to Yammer 191
15.10 Replicate MailChimp Campaign 192
15.11 Schedule MailChimp Campaign 192

15.12 Send MailChimp Campaign 193
15.13 Send Twitter Message 193
15.14 Set User's Property in Yammer 195
15.15 Subscribe to MailChimp List 196
15.16 Unsubscribe from MailChimp List 196
15.17 Update Group in Yammer 197
16. Strings activities 197
16.1 Capture All Occurrences of Text 197
16.2 Capture Text 198
16.3 Convert HTML To Text 199
16.4 Convert Number To Text 199
16.5 Convert Text Case 200
16.6 Converts Base64 to Text 200
16.7 Converts Text to Base64 201
16.8 Decode Web Text 202
16.9 Encode Web Text 203
16.10 Evaluate Mathematical Expression 204
16.11 Extract Substring 205
16.12 Format Field of Current Item 206
16.13 Get Random Number 207
16.14 Get Random String 207
16.15 Is Value Matches Regular Expression (HarePoint) 207
16.16 Join Text 208
16.17 Replace Text 209
16.18 Set Regular Expression Options 209
16.19 Split Text 210
16.20 Trim Text 211
17. Workflow activities 211
17.1 Get Workflow Association Data 211
17.2 Get Workflow Status 212
17.3 Start Workflow 212

17.4 Terminate Workflow 214
17.5 Wait Workflow To Complete 215
17.6 Wait Workflow To Complete By Name 215
Appendix A. List of activities from Free Set 217

I. Introduction

HarePoint Workflow Extensions is a software product that substantially expands the capacity to create workflows for SharePoint and in the most of cases allows avoiding custom development and coding when creating them.

The product offers a set of ready activities with a fundamentally new functionality to workflows - managing document libraries and lists, reading and writing Microsoft Office documents, document conversion to PDF and other formats, image conversion, processing and resizing, SharePoint administration, working with Active Directory, working with SQL databases and XML, executing just-in-time compiled C# or VB.NET code, PowerShell cmdlets, operating through such networking protocols as HTTP, SMTP, POP3, FTP, RSS, XMPP and Twitter, improved activities for emailing messages and a large set of activities that facilitate work with string arrays, strings and regular expressions, date and time.

At the same time, the product is fully supported by all Microsoft workflow development tools and there is no need to migrate and adapt current workflows to a new development environment, master and deploy a new development environment.

Three key features of HarePoint Workflow Extensions are as follows:

- ✓ over 200 new activities in addition to 30 built-in SharePoint activities;
- ✓ over 30 activities of Free Set, which you can use without any fee or limitation;
- ✓ full support in SharePoint Designer, Microsoft Visio[1], Microsoft Visual Studio;
- ✓ easy deployment, integration and minimum training costs.

We are confident that within an hour after deployment you will learn and be able to use new features of the product to the in full.

[1] Current version supports Microsoft Visio 2010 only; Visio 2013 integration will be added in next release

II. Deployment

1. System requirements

HarePoint Workflow Extensions is designed to operate in SharePoint Server 2013, SharePoint Server 2010, SharePoint Foundation 2013 and SharePoint Foundation 2010 (hereinafter referred to as SharePoint server) environments, and it is not compatible with SharePoint Server 2007 and Windows SharePoint Services 3.0.

The product has no special requirements to operating system and its components, RAM memory space and CPU type. The product may operate at any configurations where SharePoint server is deployed. The product requires 10 Mb of disk space on each SharePoint server.

HarePoint Workflow Extensions includes a number of supporting Client Applications and components (hereinafter referred to as the Client Applications) that can be installed on the workstations used for workflow development (hereinafter referred to as the Workstations).

Minimum workstation system requirements:

- Microsoft Windows XP Service Pack 2 (x86, x64).
- Microsoft .NET Framework v2.0.
- 10 MB of free disk space.

In addition to the minimum requirements the Client Applications have a number of special requirements:

Client Program	Description	Special requirements
Social Authorize Tool	Utility to generate authorization token for activities of Social category (see description of this activity in Section IV).	Microsoft Internet Explorer 6.0 or higher
HarePoint Visio Add-In	Library of shapes and support module for HarePoint activities in Visio 2010 (see chapter 4 of Section III).	Microsoft Visio Premium 2010 x86, x64. See also "Introduction to designing and customizing workflows": http://office.microsoft.com/en-us/sharepoint-designer-help/introduction-to-designing-and-customizing-workflows-HA101859249.aspx?CTT=3
Library for Visual Studio	HarePoint library for Visual Studio and toolbar utility (see chapter 5 of Section III).	Microsoft Visual Studio 2010 with installed project template "Sequential Workflow" for SharePoint 2010. See also "Requirements for Developing SharePoint Solutions": http://msdn.microsoft.com/en-us/library/ee231582.aspx

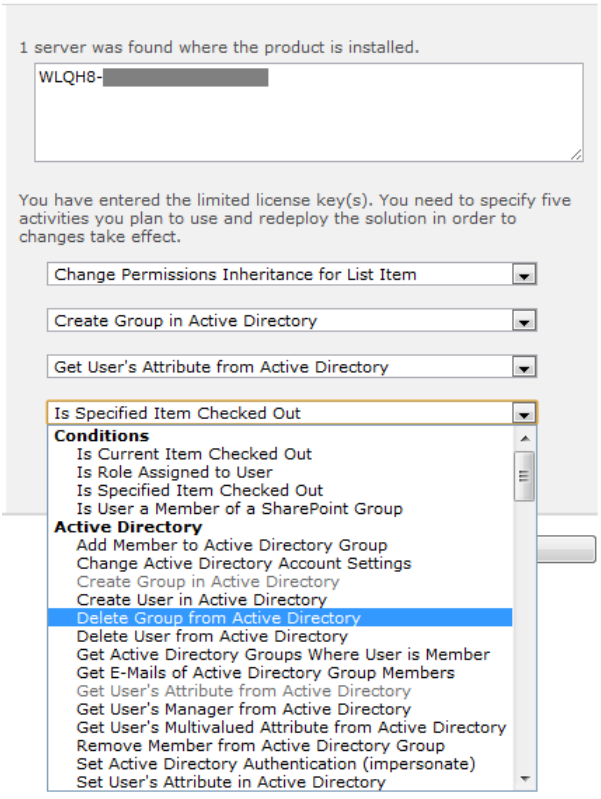
2. Licensing and trial limitations

HarePoint Workflow Extensions is licensed for a number of SharePoint servers where “Microsoft SharePoint Foundation Workflow Timer Service” is installed. The number of licenses per product should correspond to a number of servers. A list of services installed on the server can be viewed by opening System Settings tab and clicking the link “Manage servers in this farm” on Central Administration site.

A number of Workstations with installed Client Applications for workflow development aren’t accounted by product licensing.

You may use the product on the basis of one of 4 licenses:

1. *Trial License*. Provided automatically for a 30-day period to use the product in organization’s environment without any limitation by number of servers and product functionality.
2. *Free License*. This license provides free use of the product in the organization’s environment without limitation by number of servers, usage time, but with limited set of activities: a set of free activities only (see Appendix A).
3. *Limited License*. This license is purchased for a number of SharePoint servers and allows using product in the organization’s environment without limitation by usage time, however with limited set of available activities: only the set of free activities and any five paid activities at your choice.
4. *Full License*. This license is purchased for a number of SharePoint servers and allows using the product in the organization’s environment without limitation by usage time and any limitation of available activities.


1 server was found where the product is installed.
WLQH8-
You have entered the limited license key(s). You need to specify five activities you plan to use and redeploy the solution in order to changes take effect.
Change Permissions Inheritance for List Item
Create Group in Active Directory
Get User's Attribute from Active Directory
Is Specified Item Checked Out
Conditions
Is Current Item Checked Out
Is Role Assigned to User
Is Specified Item Checked Out
Is User a Member of a SharePoint Group
Active Directory
Add Member to Active Directory Group
Change Active Directory Account Settings
Create Group in Active Directory
Create User in Active Directory
Delete Group from Active Directory
Delete User from Active Directory
Get Active Directory Groups Where User is Member
Get E-Mails of Active Directory Group Members
Get User's Attribute from Active Directory
Get User's Manager from Active Directory
Get User's Multivalued Attribute from Active Directory
Remove Member from Active Directory Group
Set Active Directory Authentication (impersonate)
Set User's Attribute in Active Directory

After the first installation of the product on SharePoint server it will automatically start operating under Trial License. At the same time all the activities supplied with the products are available.

After expiration of the Trial License, the product will continue to operate under Free License. The set of available activities will be limited to the free set (see Appendix A). Execution of workflows created by using paid activities will be terminated with error and record “This workflow action requires a license” in the history list.

After purchasing a Limited License and entering registration keys on the administration page HarePoint Workflow Extensions of the Central Administration site it will be possible to choose five activities from

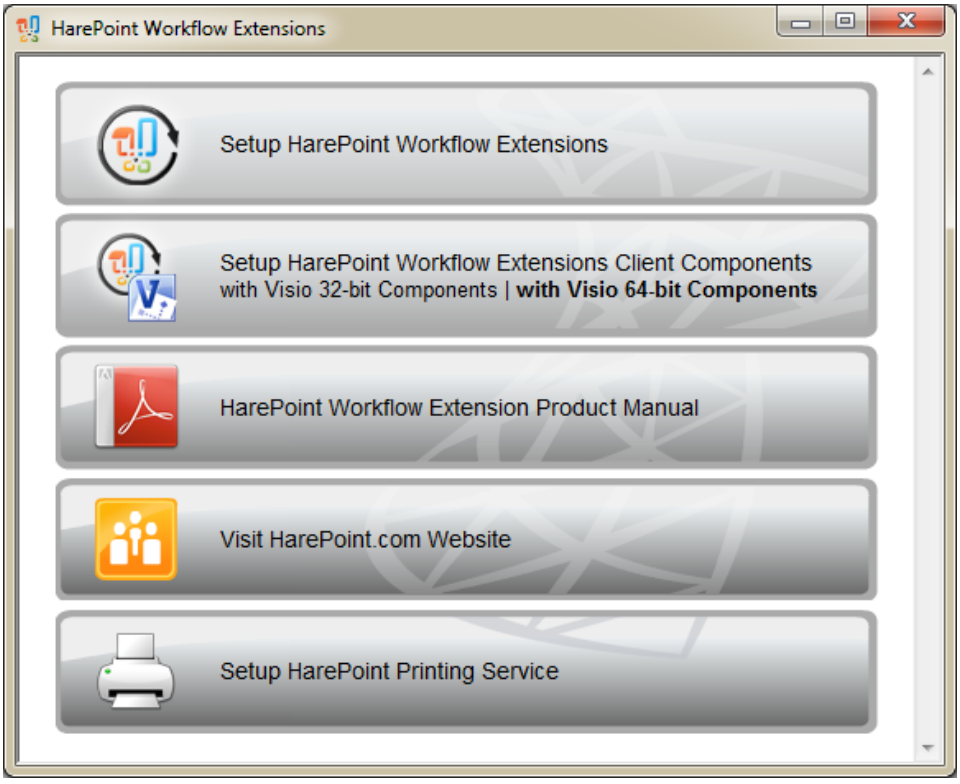
the paid list (in addition to activities from the free set). This selection can be changed later. The rest of paid activities will be inaccessible for creation of workflows, while execution of workflows created with their help will be terminated with error and record “This workflow action requires a license” in the history list.

After purchasing Full License and entering Registration Keys on the administration page HarePoint Workflow Extensions of the Central Administration site, all workflow activities will become available. All previously created workflows using the paid activities will be executed without errors.

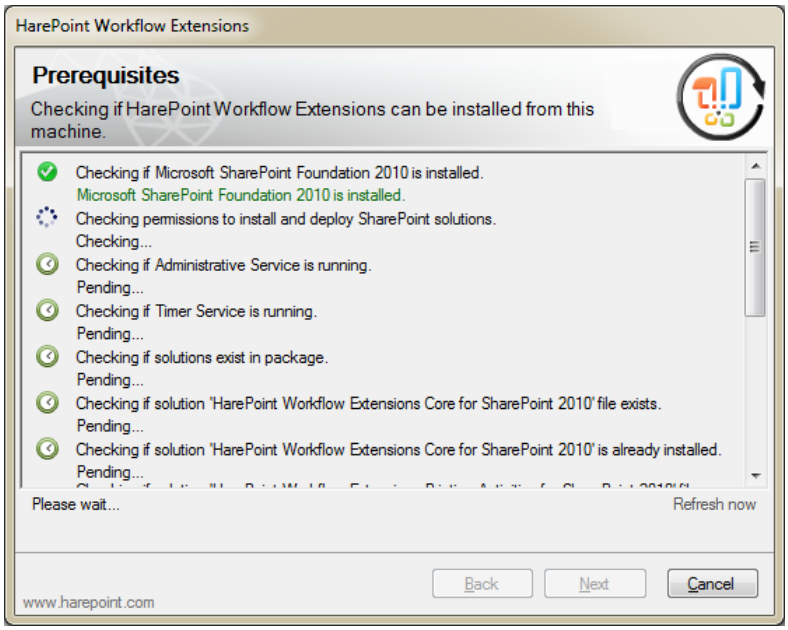
3. Installing HarePoint Workflow Extensions

To install the product you‘ll need administrator rights for SharePoint farm.

Please access any server of the SharePoint farm, unpack the product archive into a folder on the local disk. Run SETUP.EXE in the selected folder. The following window will be opened:


HarePoint Workflow Extensions
Setup HarePoint Workflow Extensions
Setup HarePoint Workflow Extensions Client Components
with Visio 32-bit Components | with Visio 64-bit Components
HarePoint Workflow Extension Product Manual
Visit HarePoint.com Website
Setup HarePoint Printing Service

Select item “Setup HarePoint Workflow Extensions”. Installation Wizard will be started. Press Next button on the wizard’s welcome window and the wizard will check if the product can be installed:

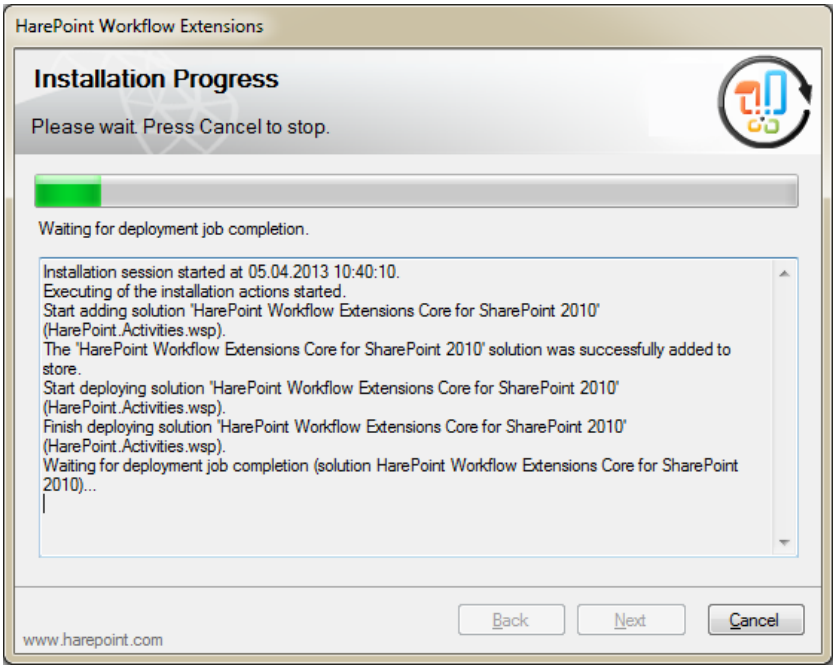

HarePoint Workflow Extensions
Prerequisites
Checking if HarePoint Workflow Extensions can be installed from this machine.
Checking if Microsoft SharePoint Foundation 2010 is installed.
Microsoft SharePoint Foundation 2010 is installed.
Checking permissions to install and deploy SharePoint solutions.
Checking...
Checking if Administrative Service is running.
Pending...
Checking if Timer Service is running.
Pending...
Checking if solutions exist in package.
Pending...
Checking if solution 'HarePoint Workflow Extensions Core for SharePoint 2010' file exists.
Pending...
Checking if solution 'HarePoint Workflow Extensions Core for SharePoint 2010' is already installed.
Pending...
Please wait...
Refresh now
Back
Next
Cancel
www.harepoint.com

Should any problems be detected, please rectify them and restart the installation wizard.

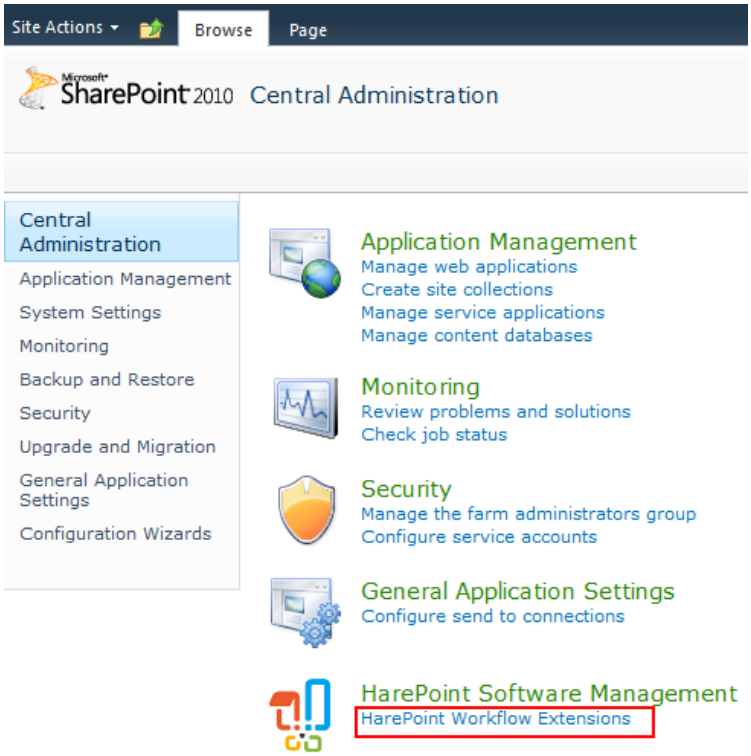
At the next step of the Wizard, you will be offered the solutions to install. In general case for successful product installation you’ll need to install both solutions offered:

- HarePoint Workflow Extensions Core – solution containing activity library.
- HarePoint Workflow Extensions Administration – solution for managing the product on the Central Administration site.
- HarePoint Printing Service Application – service application that receives printing jobs from workflow action and transfer it to HarePoint Printing Service.
- HarePoint Workflow Extensions Printing Activities – set of activities that allows printing from within a workflow.

Follow the instructions of the Wizard and complete installation of the solutions selected:

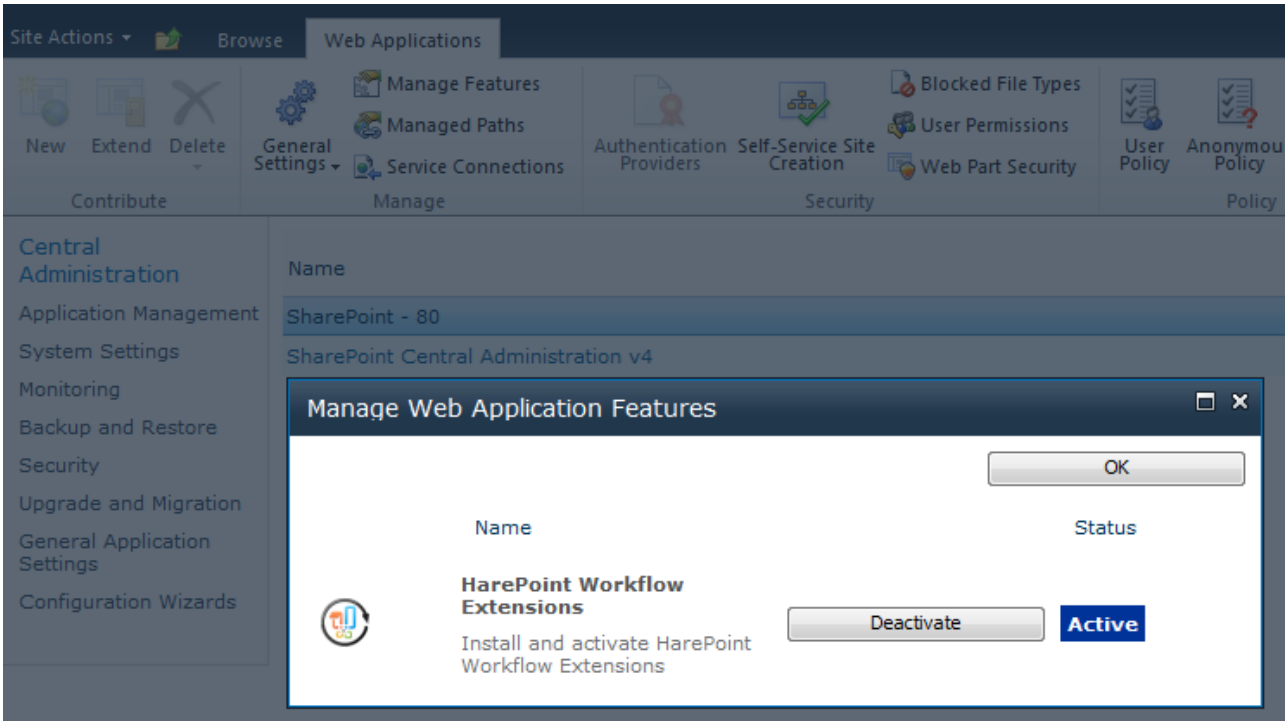

HarePoint Workflow Extensions
Installation Progress
Please wait. Press Cancel to stop.
Waiting for deployment job completion.
Installation session started at 05.04.2013 10:40:10.
Executing of the installation actions started.
Start adding solution 'HarePoint Workflow Extensions Core for SharePoint 2010' (HarePoint.Activities.wsp).
The 'HarePoint Workflow Extensions Core for SharePoint 2010' solution was successfully added to store.
Start deploying solution 'HarePoint Workflow Extensions Core for SharePoint 2010' (HarePoint.Activities.wsp).
Finish deploying solution 'HarePoint Workflow Extensions Core for SharePoint 2010' (HarePoint.Activities.wsp).
Waiting for deployment job completion (solution HarePoint Workflow Extensions Core for SharePoint 2010)...
Back
Next
Cancel
www.harepoint.com

Open Central Administration site. Open the “HarePoint Workflow Extensions” link in HarePoint Software Management section:



If you have product license, please enter the license code in “Licenses” section. This section also describes the type of your current license and in case of Trial License, the number of days left to its expiration.

Please be advised that deactivation of “HarePoint Workflow Extensions” in the “Manage Features” section of the web application will not result in deactivation of product features:

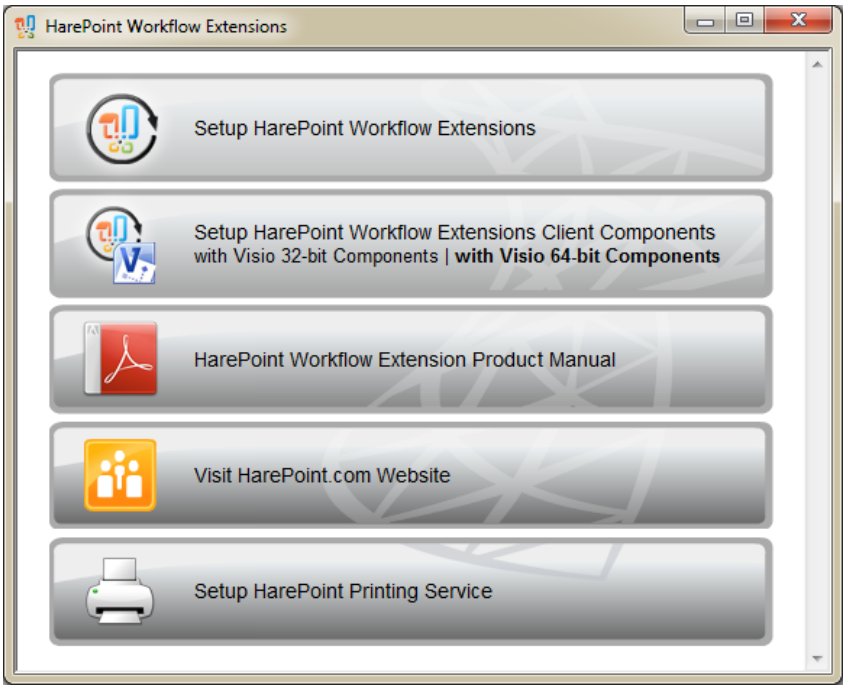


Manual on forced deactivation of full or partial functionality of the products for the site is given in chapter 6, Section III of this Manual.

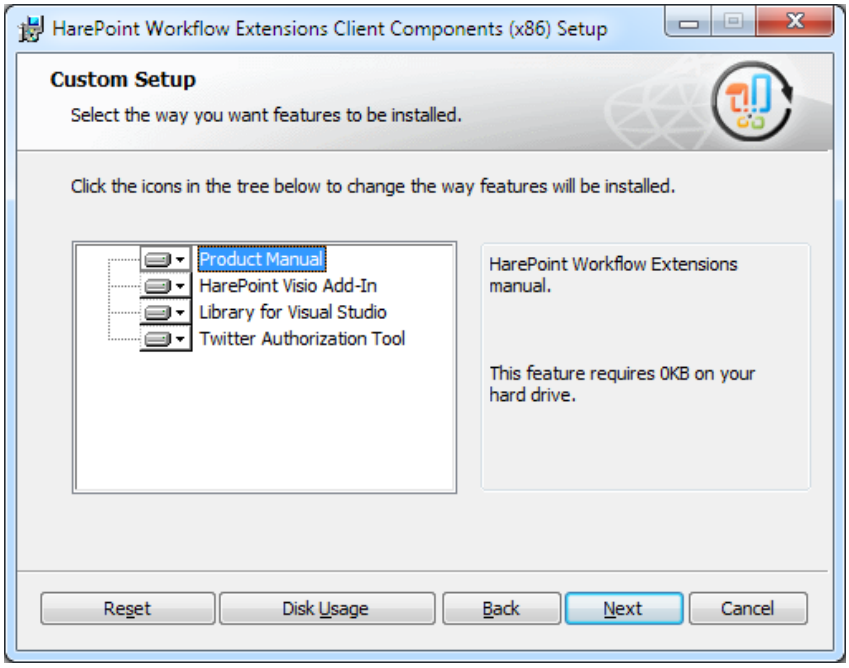
4. Installing Client Applications

Installation of Client Applications is not mandatory for workflow development using the product activities (see chapter 1 of this Section and chapter 1 of Section III).

Enter the Workstation; unpack the product archive into a folder on the local disk. Run SETUP.EXE in the selected folder. The following window will be opened:


HarePoint Workflow Extensions
Setup HarePoint Workflow Extensions
Setup HarePoint Workflow Extensions Client Components
with Visio 32-bit Components | with Visio 64-bit Components
HarePoint Workflow Extension Product Manual
Visit HarePoint.com Website
Setup HarePoint Printing Service

Select the “Setup HarePoint Workflow Extensions Client Components” item. Installation Wizard will be launched. Follow instructions of the Wizard to select the required Client Applications:


HarePoint Workflow Extensions Client Components (x86) Setup
Custom Setup
Select the way you want features to be installed.
Click the icons in the tree below to change the way features will be installed.
Product Manual
HarePoint Visio Add-In
Library for Visual Studio
Twitter Authorization Tool
HarePoint Workflow Extensions manual.
This feature requires 0KB on your hard drive.
Reset
Disk Usage
Back
Next
Cancel

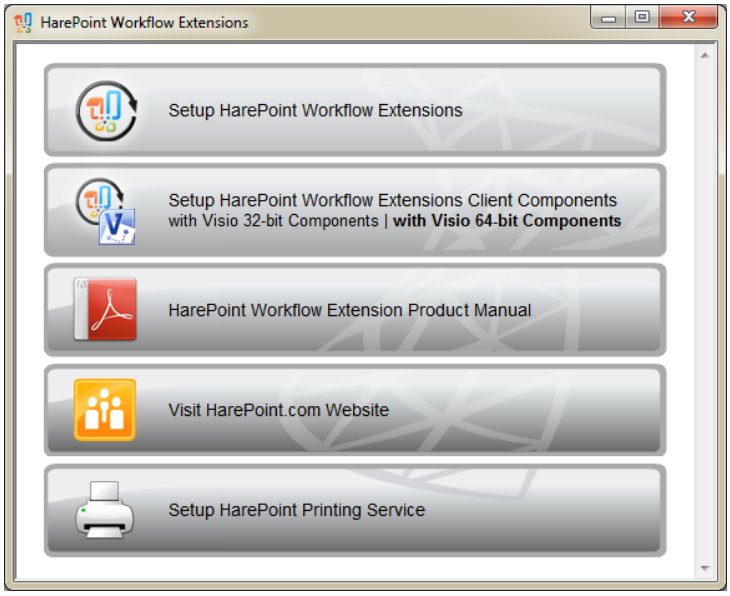
Then follow instructions of the Wizard and complete the installation.

The operation with Client Application is described in Section III of this manual.

5. How to repair, upgrade or remove product

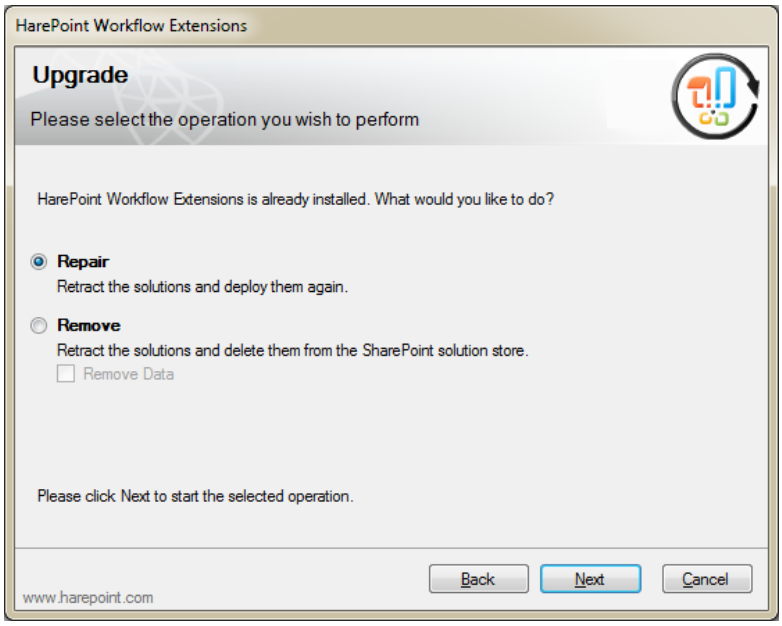
To repair, upgrade or remove the product you'll need administrator rights of SharePoint farm.

Please enter any server of the SharePoint farm, unpack the product archive into a folder on the local disk. Run SETUP.EXE in the selected folder. The following window will be opened:

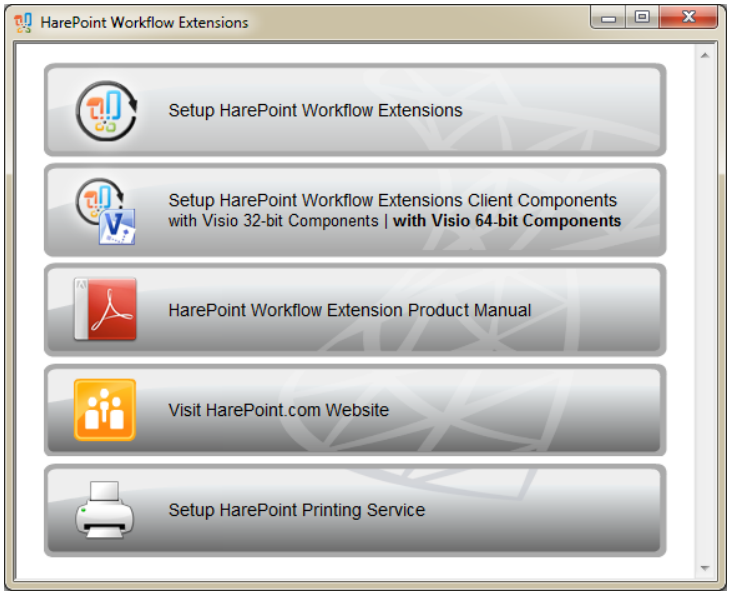

HarePoint Workflow Extensions
Setup HarePoint Workflow Extensions
Setup HarePoint Workflow Extensions Client Components
with Visio 32-bit Components | with Visio 64-bit Components
HarePoint Workflow Extension Product Manual
Visit HarePoint.com Website
Setup HarePoint Printing Service

Please select "Setup HarePoint Workflow Extensions" item. Installation Wizard will be launched. Following instructions of the Wizard, pass system check and solution selection steps similar to those during the product installation (see chapter 3).

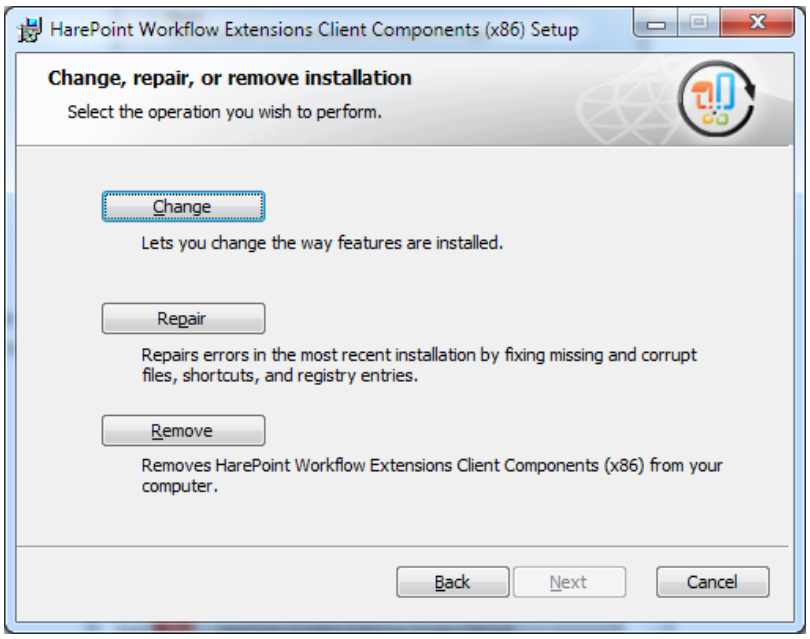
Upon completing these steps the Wizard will ask to repair, upgrade or remove the product components. Select the required type of operation and follow the further instructions of the Wizard.


HarePoint Workflow Extensions
Upgrade
Please select the operation you wish to perform
HarePoint Workflow Extensions is already installed. What would you like to do?
Repair
Retract the solutions and deploy them again.
Remove
Retract the solutions and delete them from the SharePoint solution store.
Remove Data
Please click Next to start the selected operation.
www.harepoint.com
Back
Next
Cancel

To repair, upgrade or remove a Client Application, please enter a Workstation, unpack the product archive into a folder on the local disk. Run SETUP.EXE in the selected folder. The following window will be opened:


HarePoint Workflow Extensions
Setup HarePoint Workflow Extensions
Setup HarePoint Workflow Extensions Client Components
with Visio 32-bit Components | with Visio 64-bit Components
HarePoint Workflow Extension Product Manual
Visit HarePoint.com Website
Setup HarePoint Printing Service

Select “Setup HarePoint Workflow Extensions Client Components” item. Installation Wizard will be launched. Following instructions of the Wizard, select the required Client Applications and type of operation:


HarePoint Workflow Extensions Client Components (x86) Setup
Change, repair, or remove installation
Select the operation you wish to perform.
Change
Lets you change the way features are installed.
Repair
Repairs errors in the most recent installation by fixing missing and corrupt files, shortcuts, and registry entries.
Remove
Removes HarePoint Workflow Extensions Client Components (x86) from your computer.
Back
Next
Cancel

Follow instructions of the Wizard and complete installation.

III. Product Overview

1. Getting started

Microsoft offers three tools that create workflows for SharePoint. Two of them, SharePoint Designer and Microsoft Visio, are intended for no-code workflow automation. A third one, Microsoft Visual Studio, includes more functionality but requires quite complex programming and deep knowledge of SharePoint architecture.

HarePoint Workflow Extensions can be successfully used to develop workflows in Visual Studio, but this product's main goal is to avoid programming as this requires not only deep knowledge but also ten times more time for workflow development, debugging and support.

If you are new to workflow development in SharePoint Designer and Microsoft Visio, before you further read this material we recommend you the article "Introduction to designing and customizing workflows" at http://office.microsoft.com/en-us/sharepoint-designer-help/introduction-to-designing-and-customizing-workflows-HA101859249.aspx?CTT=3

Please note that the product is made up of five components:

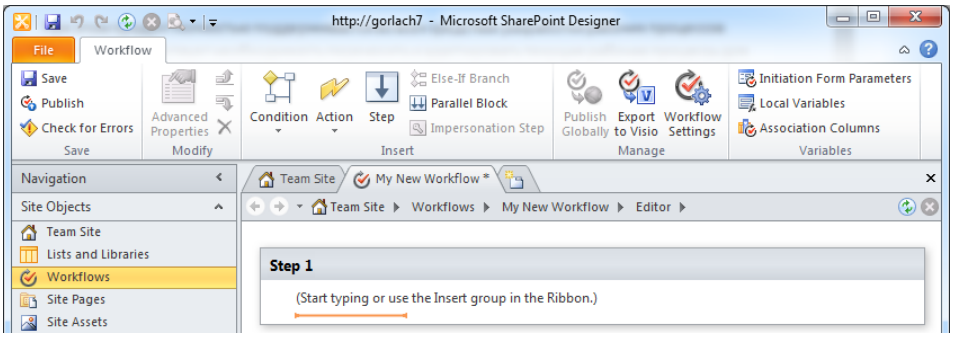
- The library of the activities. Installed on servers of SharePoint farm.
- The administration module. Installed on the Central Administration site of SharePoint farm.
- A plug-in for Microsoft Visio[2]. Installed on the work station where it requires use of HarePoint Workflow Extensions activities in Microsoft Visio.
- The library for Visual Studio. Installed on the work station where it requires use of HarePoint Workflow Extensions activities in Visual Visio.
- Utility for Social networks. Installed on the work station, required for generation of an authorization token.

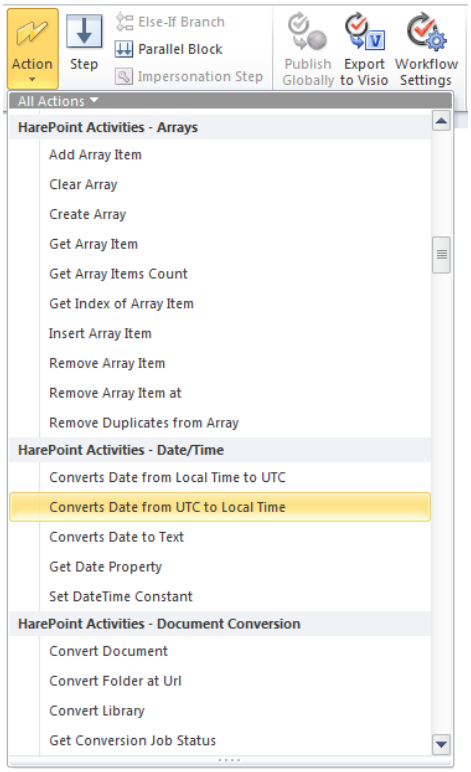
The product files do not have to be installed on the work station to develop workflows only in SharePoint Designer without the use of Microsoft Visio и Visual Studio. The exception is the utility for Social networks required to generate an authorization token for the activities of Social category (see description in Section IV).

[2] Current version supports Microsoft Visio 2010 only; Visio 2013 integration will be added in next release

2. Creating workflows in SharePoint Designer

Within SharePoint Designer, get connected to the site where the product is installed, create a new workflow or open the existing workflow:





Click “Action” on the control panel and scroll the “All Actions” list that opens. If you see categories of activities the titles of which start with “HarePoint Activities” – congratulations, as everything has been successfully installed and you can start creating workflows.

Please note that apart from activities the product includes a few conditions. They are available in the drop-down list of Conditions and have the names of this kind:

Is Item Field Matched to Regular Expression (HarePoint)

Working with HarePoint activities and conditions is not different from working with in-built SharePoint activities and conditions. A complete user manual for activities and conditions is provided in Section IV of this user manual.

If you are new to workflow development in SharePoint Designer, please read the article referred to in “Getting Started”.

Please note that a number of activities and conditions are context-sensitive and sometimes you may not find them in the menu. For example, the “Create Document from File” activity is available only when creating a workflow for the Document Library and it won’t be listed in the menu when a workflow for the Calendar or the Task List is created.

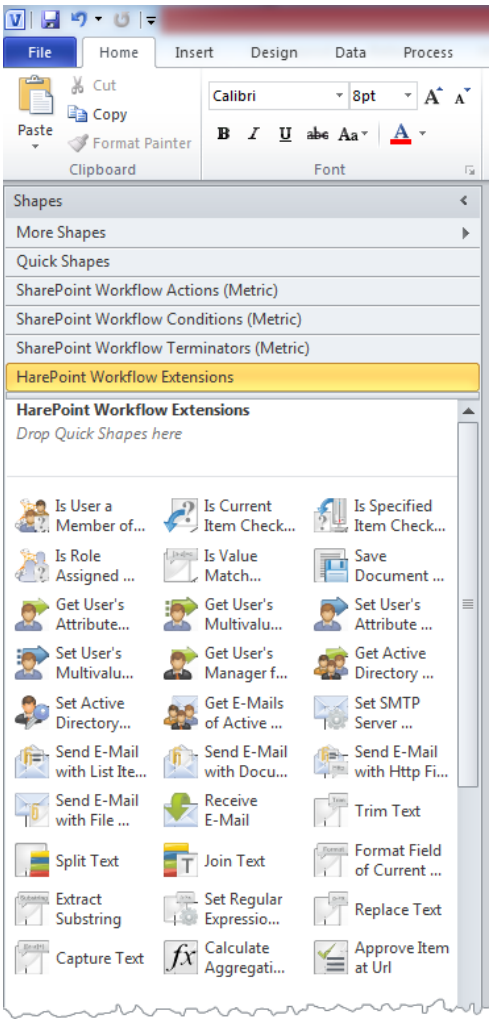
3. Designing workflows in Microsoft Visio

Microsoft Visio allows easily designing large and complex workflows. You cannot set parameters for activities in Visio. However, it visually represents the workflow algorithm.

Note: current version supports Microsoft Visio 2010 only; Visio 2013 integration will be added in next release.

If you are new to designing workflows in Visio, please read the following articles:

- Introduction to designing and customizing workflows: http://office.microsoft.com/en-us/sharepoint-designer-help/introduction-to-designing-and-customizing-workflows-HA101859249.aspx?CTT=3
- Create, import, and export SharePoint workflows in Visio: http://office.microsoft.com/en-us/visio-help/create-import-and-export-sharepoint-workflows-in-visio-HA101888007.aspx



To design workflows in Visio, Microsoft Visio Premium is required. Besides, HarePoint Visio Plug-In should be installed on the computer for full support of HarePoint Workflow Extensions activities (see Section II of this manual).

Select File -> Options in Microsoft Visio and go to “Add-Ins” in the “Visio Options” window that opens. There should be “HarePoint Visio Add-In” in the “Active Application Add-Ins” section.

When a new document is being created from a “Microsoft SharePoint Workflow” template or a SharePoint workflow is being imported, “HarePoint Workflow Extensions” tab will appear in Shapes.

Using HarePoint activities and conditions in Microsoft Visio is not different from using in-built SharePoint Workflow activities and conditions.

Please note that HarePoint Visio Plug-In is used for import and export of a workflow file. The plug-in is not used to save files of other types. When there is some failure during import or export of workflow files, please disable the plug-in in the “Visio Options” menu.

Please always back up workflow files.

4. Creating workflows in Visual Studio

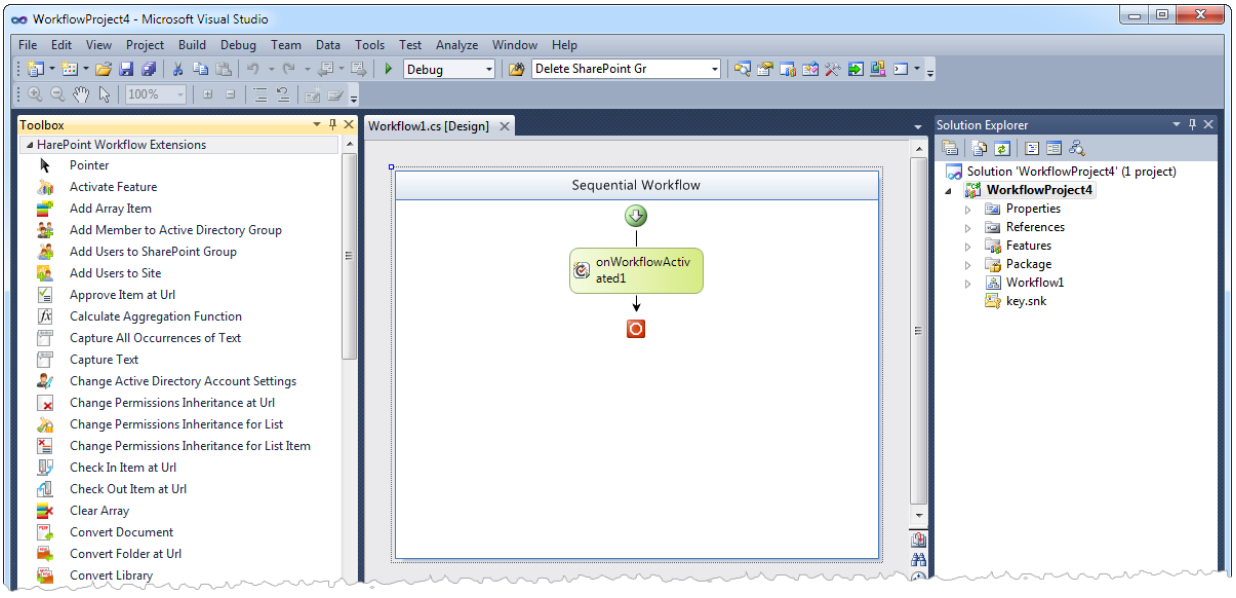
Creating workflows in Visual Studio is rather complex and required deep knowledge of SharePoint architecture. Whenever possible, it's better to confine with "Execute Custom Code" activity or with development of a custom activity rather than create a workflow in Visual Studio.

If you are new to developing workflows in Visual Studio, please first read the following articles:

- Requirements for Developing SharePoint Solutions: http://msdn.microsoft.com/en-us/library/ee231582.aspx
- Creating SharePoint 2010 Sequential Workflows in Visual Studio 2010: http://msdn.microsoft.com/en-us/library/gg265727.aspx

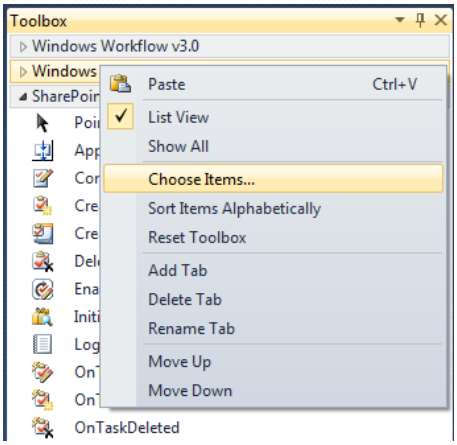
When developing a workflow in Visual Studio, you can use HarePoint activities and in-built SharePoint activities alike. For this, all components should be installed on the computer as required by the article "Requirements for Developing SharePoint Solutions" as well as HarePoint libraries for Visual Studio (see Section II of this manual).

Select "Add HarePoint toolbox to Visual Studio 2010" in Start -> Programs. A console application will be launched to add the "HarePoint Workflow Extensions" toolbox for all languages and all projects created on the basis of "Sequential Workflow" and "State Machine Workflow" templates for SharePoint 2010.



The toolbox can be removed through the context menu, "Delete Tab" in the toolbox, or by running "Remove HarePoint toolbox from Visual Studio 2010" in Start -> Programs.

A toolbox can also be added manually. For this, select “View \ Toolbox” in the project based on the “Sequential Workflow” template created as described in the article referred to above. Click the right button of the mouse in the work area in the Toolbox window that opens and select “Choose Items…”.



Click “Browse…” in the “.NET Framework Components” tab of the window that opens to select the *HarePoint.Activities.Dll* file in the directory selected for installation of the HarePoint library for Visual Studio. Then HarePoint activities appear in the list.

After you press OK and close the window, a link to HarePoint.Activities will appear in the References section of the project, the “General” tab with HarePoint Workflow Extensions activities will appear in the Toolbox window.

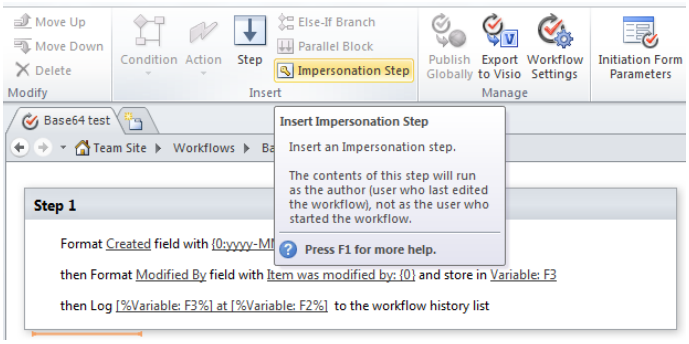
After components are added manually, they can be renamed manually so that their names will be more readable and look like in SharePoint Designer and sorted by name by selecting the “Sort Items Alphabetically” item in the context menu. Please note that when components are added by a command “Add HarePoint toolbox to Visual Studio 2010”, they will be automatically renamed and sorted.

Using HarePoint activities in Visual Studio is not different from using in-built SharePoint activities.

5. Security issues

HarePoint activities are performed in the context of the user who initiated a workflow. If we have created a workflow that automatically runs when a new document is added to a library, this workflow will always run in the context of the user who added the document.

In this case, a number of activities such as “Create User in Active Directory” or “Add User to SharePoint Group” require adequate rights and permissions. The user who has the permission for adding documents to a library may not have rights and permissions for creating a user in Active Directory or adding a user in the SharePoint group and the workflow will be run with errors.

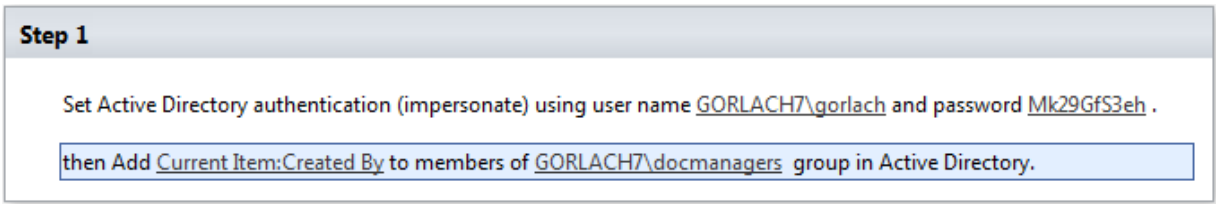


Such mistakes can be avoided in two ways.

A general method is to move such activities to the Impersonation Step, which is run in the context of the workflow author (or last user who edited the workflow).

The second method is available only for activities working with the Active Directory.

It involves adding the “Set Active Directory Authentication (impersonate)” activity which saves the login and password for access to the Active Directory in the current workflow step.


Step 1
Set Active Directory authentication (impersonate) using user name GORLACH7\gorlach and password Mk29GfS3eh .
then Add Current Item:Created By to members of GORLACH7\docmanagers group in Active Directory.

All HarePoint activities working with the Active Directory check if the current workflow step has the above “Set Active Directory Authentication (impersonate)” activity and, if it does, use access parameters set there. Otherwise, they work with the Active Directory in the context of the user who initiated workflow launch or the workflow author if they were moved to the Impersonation Step.

Apart from “Set Active Directory Authentication (impersonate)”, the HarePoint library has a number of activities which also require a password. These are activities for FTP, receiving messages through POP3 and sending email through SMTP. In all cases, the password is stored as open text in the workflow XOML file.

This means that under certain conditions the password can be obtained by third parties through SharePoint Designer or even a browser through a link of this kind: http://sharepoint/Workflows/workflowname/workflowname.xoml

Setting up security for SharePoint is outside the scope of this article. If you have questions, please see the official user manual at http://technet.microsoft.com/en-us/library/ee428324.aspx

Please note that you can disable some HarePoint activities. See chapter 6 of this Section.

6. How to disable certain HarePoint activities

In some cases you may need to disable certain HarePoint activities. What you need to do is open the *web.config* file on the relevant server and add a string as shown in the example below (italicized) to the authorizedTypes section:

```
<System.Workflow.ComponentModel.WorkflowCompiler>
  <authorizedTypes>
    ....
    <authorizedType Assembly="HarePoint.Activities, Version=2.0.0.0, Culture=neutral,
            PublicKeyToken=73210640c0bcfb88"
            Namespace="HarePoint.Activities" TypeName="*" Authorized="True" />
    <authorizedType Assembly="HarePoint.Activities, Version=2.0.0.0, Culture=neutral, PublicKeyToken=73210640c0bcfb88"
            Namespace="HarePoint.Activities" TypeName="Actions.FormatField" Authorized="False" />
  </authorizedTypes>
```

```
</System.Workflow.ComponentModel.WorkflowCompiler>
```

After IIS is restarted, all workflows with the "Format Field of Current Item" activity (HarePoint.Activities.Actions.FormatField) will terminate with an error - “Workflow was canceled by System Account”. In response to an attempt to add a new workflow with this activity, SharePoint Designer will show an error message *“(0, 0) Type HarePoint.Activities.Actions.FormatField, HarePoint.Activities, Version=2.0.0.0, Culture=neutral, PublicKeyToken=73210640c0bcfb88 is not marked as authorized in the application configuration file.)*”.

To remove an activity from the list of those available for SharePoint Designer, open the file “HarePointActivities.Actions” in the templates directory (such as: "C:\Program Files\Common Files\Microsoft Shared\Web Server Extensions\14\TEMPLATE\1033\Workflow\") and remove the following block from it:

```
<Action Name="Format Field of Current Item"
  ClassName="HarePoint.Activities.Actions.FormatField"
  Assembly="HarePoint.Activities, Version=2.0.0.0, Culture=neutral, PublicKeyToken=73210640c0bcfb88"
  AppliesTo="all"
  Category="HarePoint Activities - Strings">
 <RuleDesigner Sentence="Format %1 field with %2 and store in %3">
  <FieldBind Field="Input" DesignerType="FieldNames" Text="this" Id="1"/>
  <FieldBind Field="Format" DesignerType="TextArea" Text="this format" Id="2"/>
  <FieldBind Field="OutputText" DesignerType="ParameterNames" Text="FormattedText" Id="3"/>
 </RuleDesigner>
 <Parameters>
  <Parameter Name="__Context"
       Type="Microsoft.Sharepoint.WorkflowActions.WorkflowContext, Microsoft.SharePoint.WorkflowActions"
       Direction="In"/>
  <Parameter Name="__ListId" Type="System.String, mscorlib" Direction="In" />
  <Parameter Name="__ListItem" Type="System.Int32, mscorlib" Direction="In" />
  <Parameter Name="Input" Type="System.String, mscorlib" Direction="In"
       Description="Field name of current item"/>
  <Parameter Name="Format" Type="System.String, mscorlib" Direction="In" Description="Format string"/>
  <Parameter Name="OutputText" Type="System.String, mscorlib" Direction="Out" Description="Output text"/>
 </Parameters>
</Action>
```

After IIS is restarted, there’ll be no activity in the list of activities available in SharePoint Designer for all sites on the server.

IV. HarePoint Activities and Conditions

This section contains manual on activities and conditions of the product. How the action is initially displayed in a workflow step (appears in SharePoint Designer upon adding to the workflow) is given for each activity or condition after its name. Some descriptions contain a paragraph of examples with sample workflows and how they are displayed in SharePoint Designer unless otherwise specified.

1. Active Directory activities

1.1 Add Member to Active Directory Group

Add current user to members of specified group in Active Directory.

This workflow action is used to add one or several users or groups to specified Active Directory security group.

Note: In order to specify AD credentials, add the "Set Active Directory Authentication" activity to the same workflow step first. Otherwise, AD will be accessed under context of the user who started the workflow or the workflow author context in the case of impersonation step.

Parameters:

current user - Logon name of the user or group: DOMAIN\user or user@domain.com. Leave the field empty to add the current user.
To add **multple users**, provide their logon names semicolon-delimited: mapilab\test1;mapilab\test2;mapilab\test3
specified - Logon name of the group: DOMAIN\group or group@domain.com.

1.2 Change Active Directory Account Settings

Change logon name account settings: enable, change password at next logon: keep existing, password never expires: keep existing, account expires: yes on this date

This workflow action is used to enable or disable user account and change its settings.

See Also: "Create User in Active Directory" activity.

Note: In order to specify AD credentials, add the "Set Active Directory Authentication (impersonate)" activity to the same workflow step first. Otherwise, AD will be accessed under context of the user who started the workflow or workflow author context in the case of impersonation step.

Parameters:

logon name - Logon name of the user: DOMAIN\user or user@domain.com.
enable/disable - Enable or disable user account.
change password - User must change password at next logon: yes/no or keep existing settings
password never expires - Password never expires: yes/no or keep existing settings.
account expires: yes - Select if the account expires: yes/no or keep existing settings. If Yes is selected, please specify the expiry date in the next parameter.
on this date - specify the expiry date for an account, if you selected that this account expires (previous parameter set to Yes).

1.3 Change Local User Account Settings on Server

Change logon name account settings on this server: enable, change password at next logon: keep existing, password never expires: keep existing.

This workflow action changes local user account on a specified server.

Important Note: *'Set Active Directory authentication'* action is required prior using this action to specify the administrator account that will be used to change properties of a local account

Parameters:

logon name - local account logon name
this server - NetBIOS name, DNS name or IP address. **Important note:** if you need to change a user on a local machine, use *localhost* as a name!
enable - enable or disable this local user account
change password at next logon: keep existing

Options available: Yes, No, Keep existing
password never expires: keep existing
Options available: Yes, No, Keep existing

Examples:

Set Active Directory authentication (impersonate) using name COMPANY\Administrator and password 111.

Then Set Variable: logon name to test-n1

then Create new user on 192.168.2.16 server with Variable: logon name logon name, John Smith full name, Test user description, 123 password.

then Change Variable:logon name account settings on 192.168.2.16 : enable, change password at next logon: yes , password never expires: yes.

1.4 Create Group in Active Directory

Create new group in this container with specified group name.

This workflow action will create new group in specified Active Directory container.

Note: In order to specify AD credentials, add the "Set Active Directory Authentication (impersonate)" activity to the same workflow step first. Otherwise, AD will be accessed under context of the user who started the workflow or workflow author context in the case of impersonation step.

Parameters:

this - Distinguished name of the Active Directory container: OU=Test,DC=domain,DC=com. Use Active Directory Explorer to get correct path: http://technet.microsoft.com/en-us/sysinternals/bb963907
specified - Logon name of the group: DOMAIN\group or group@domain.com.

1.5 Create Local User Account on Server

Create new user on this server with specified logon name, this full name, this description, this password.

this workflow action creates a local user account on a specified server.

Important Note: '*Set Active Directory authentication*' action is required prior using this action to specify the administrator account that will be used to create a local account

Parameters:

this server - NetBIOS name, DNS name or IP address. **Important note:** if you need to create a user on a local machine, use *localhost* as a name!
specified logon name - the logon name for the account
this full name - (optional) the account full name
this description - (optional) the description for the account
this password - the password for account

Examples:

Set Active Directory authentication (impersonate) using name COMPANY\Administrator and password 111.

Then Set Variable: logon name to test-n1

then Create new user on 192.168.2.16 server with Variable: logon name logon name, John Smith full name, Test user description, 123 password.

1.6 Create Mail Contact in Active Directory

Create new contact in this container with specified name, this display name, this alias, this external e-mail, mail enable: yes.

This workflow action creates a new mail contact with specified parameters.

Parameters:

this container - specify the distinguished name or canonical name of the container where the contact will be created.
with specified name - specify the name for the contact object in AD
this display name - specify the name that will be displayed to the user
this alias - (for mail-enabled contacts only) the internal e-mail address of the contact

this external e-mail - external e-mail address (outside of organization)
mail enable: yes - enable (or disable) receiving e-mails to this contact via Microsoft Outlook and Microsoft Exchange

1.7 Create User in Active Directory

Create new user in this container with specified logon name, this first name, this last name, this password.

This workflow action will create new user account in specified Active Directory container. Created account will be enabled, you can disable account and change its settings with "Change Active Directory Account Settings" activity.

Note: In order to specify AD credentials, add the "Set Active Directory Authentication (impersonate)" activity to the same workflow step first. Otherwise, AD will be accessed under context of the user who started the workflow or workflow author context in the case of impersonation step.

Parameters:

this container - Distinguished name of the Active Directory container, like OU=Test,DC=domain,DC=com.
specified - User's logon name. Domain name in the user's logon name is ignored (correct syntax is DOMAIN\username or username@domain).
this first name - User's first name.
this last name - User's last name.
this password - User's password. Password may be blank. Use "Change Active Directory Account Settings" activity to ask user to change password at first logon.

1.8 Delete Group from Active Directory

Delete group with this group name from Active Directory.

This workflow action will delete specified security group from Active Directory.

Note: In order to specify AD credentials, add the "Set Active Directory Authentication (impersonate)" activity to the same workflow step first. Otherwise, AD will be accessed under context of the user who started the workflow or workflow author context in the case of impersonation step.

Parameters:

this - Logon name of the group: DOMAIN\group or group@domain.com.

1.9 Delete Local User Account from Server

Delete user with this logon name from this server.

This workflow action deletes the specified local user account from the server.

Important Note: *'Set Active Directory authentication'* action is required prior using this action to specify the administrator account that will be used to create a local account

Parameters:

this logon name - local account logon name

this server - NetBIOS name, DNS name or IP address.

Important Note: if you need to delete a user on a local machine, use localhost as a name!

Examples:

Set Active Directory authentication (impersonate) using name COMPANY\Administrator and password 111.

Then Set Variable: logon name to test-n1

then Delete user with Variable: logon name logon name from 192.168.2.16

1.10 Delete User from Active Directory

Delete user with this logon name from Active Directory.

This workflow action will delete specified user from Active Directory.

Note: In order to specify AD credentials, add the "Set Active Directory Authentication (impersonate)" activity to the same workflow step first. Otherwise, AD will be accessed under context of the user who started the workflow or workflow author context in the case of impersonation step.

Parameters:

this - Logon name of the user: DOMAIN\user or user@domain.com.

1.11 Find User in Active Directory

Find current user in Active Directory and store logon name in Variable: variable

This workflow action is used to find a SharePoint user in Active Directory and store his logon name in the output variable. Action returns an empty string if the user is not found in Active Directory.

Note: In order to specify AD credentials, add the "Set Active Directory Authentication (impersonate)" activity to the same workflow step first. Otherwise, AD will be accessed under context of the user who started the workflow, or workflow author context in the case of impersonation step.

Parameters:

current user - Name of SharePoint user.
Variable: variable - Output string variable with a logon name like DOMAIN\username. Empty string if user is not found in Active Directory.

1.12 Find User in Active Directory by Query

Find user in Active Directory by simple query using this query and store logon name in Variable: variable.

This workflow action is used to find user or group in Active Directory by specified query. You can use two type of queries.

First, the format in style of INI file:

<AD_attribute_name1>=<value1>
<AD_attribute_name2>=<value2>

For example:

displayName=John Smith

Second, is LDAP format. For example:

(&(objectCategory=user)(objectClass=user)(displayName=John Smith)(department=Marketing))

Parameters:

simple query - One of the following query types: simple query (for .INI-style query), LDAP filter (for LDAP query).
this query - String with the query.

Variable: variable - Output string variable with a logon name like DOMAIN\username. Empty string if user is not found in Active Directory.

1.13 Get Active Directory Groups Where User is Member

Get list of the Active Directory groups where user current user is member. Store list of these groups in Variable: GroupsArray, success of the searching in Variable: UserFound.

This workflow action will return the list of groups where specified user is a member.

Note: In order to specify AD credentials, add the "Set Active Directory Authentication (impersonate)" activity to the same workflow step first. Otherwise, AD will be accessed under context of the user who started the workflow or workflow author context in the case of impersonation step.

Parameters:

current user - The name of Active Directory user for reading names of his groups (like DOMAIN\Account). Leave the field empty to use current user.

Variable: GroupsArray - Output array variable with the list of groups. Use "Join Text" or array activities to convert this array to string.

Variable: UserFound - Output boolean variable, *true* indicates that user was found in Active Directory.

Examples:

Get list of the Active Directory groups where user current user is member. Store list of these groups in Variable: GroupsArray, success of searching in Variable: UserFound.

then Get the number of items in the Variable: GroupsArray and store in Variable: groupsCount.

then Log [%Workflow Context: Current User%] is the member of [%Variable: groupsCount%] groups to the workflow history list

1.14 Get E-Mails of Active Directory Group Members

Get user e-mails from specified group. Expand nested groups: No. Store result in Variable: EmailsArray.

This workflow action will return Active Directory group member's email addresses to array variable.

Use "Join Text" or array activities to get string values.

Note: In order to specify AD credentials, add the "Set Active Directory Authentication (impersonate)" activity to the same workflow step first. Otherwise, AD will be accessed under context of the user who started the workflow or workflow author context in the case of impersonation step.

Parameters:

specified group - Name of the group from which we want to read member e-mails (like: DOMAIN\group).
expand nested groups - Include or not members of nested groups.
Variable: EmailsArray - Output array variable with the list of the e-mails of Active Directory group members.

1.15 Get User's Attribute from Active Directory

Get this attribute of current user from Active Directory. Store attribute value in Variable: variable1, success of searching in Variable: variable2.

This workflow action will return a value of Active Directory user specified attribute.

Note: In order to specify AD credentials, add the "Set Active Directory Authentication (impersonate)" activity to the same workflow step first. Otherwise, AD will be accessed under context of the user who started the workflow or workflow author context in the case of impersonation step.

See Also: "Get User's Multivalued Attribute from Active Directory" action.

Parameters:

this attribute - Name of the attribute.
current user - Name of the user from which we want to read attribute value (like DOMAIN\Account). Leave the field empty to use current user.
Variable: variable1 - Output variable for the found attribute value.
Variable: variable2 - Output boolean variable, *true* indicates that user (not attribute!) was found in Active Directory.

1.16 Get User's Manager from Active Directory

Get manager of current user from Active Directory. Store manager's display name in Variable: variable1, success of the searching in Variable: variable2.

This workflow action is used to get a manager's display name for specified Active Directory user.

Note: In order to specify AD credentials, add the "Set Active Directory Authentication (impersonate)" activity to the same workflow step first. Otherwise, AD will be accessed under context of the user who started the workflow or workflow author context in the case of impersonation step.

Parameters:

current user - The name of Active Directory user for reading his manager's name (like DOMAIN\Account). Leave the field empty to use current user.
Variable: variable1 - Output variable for resultant manager display name.
Variable: variable2 - Output boolean variable, *true* indicates that user (not manager or display name) was found in Active Directory.

1.17 Get User's Multivalued Attribute from Active Directory

Get multi-valued attribute this attribute of current user from Active Directory. Store attribute value in Variable: ArrayVariable, success of the searching in Variable: variable.

This workflow action is used to get multi-valued attribute of specified Active Directory user.

Note: This action returns array variable. You can use "Get Array Items Count", "Get Array Item" or "Join Text" actions to get items from output array.

See Also: "Get User's Attribute from Active Directory" action.

Parameters:

this attribute - Name of attribute.
current user - User logon name (like DOMAIN\Account) or current user by default.
Variable: ArrayVariable - Output array variable with attribute values.
Variable: variable - Output boolean variable, indicates that user was found in Active Directory.

1.18 Is User a Member of an Active Directory Group (HarePoint)

If current user is a member of Active Directory group this group (use this user and this password to connect Active Directory)

This condition returns true if specified user is the member of specified Active Directory group.

You can specify optional credentials for Active Directory access. Otherwise, AD will be accessed under context of the user who started the workflow or workflow author context in the case of impersonation step.

Note: The password will be stored as a plain text in the workflow XOML file and may be accessed by the URL like *http://sharepoint/Workflows/workflowname/workflowname.xoml* or by SharePoint Designer.

Parameters:

current user - User logon name, like DOMAIN\user.
this group - Group logon name, like DOMAIN\group.
this user - User name to be used for Active Directory access (like DOMAIN\Account). Optional.
this password - Password to be used for Active Directory access. Optional.

1.19 Move User into Organizational Unit

Move Active Directory user with this logon name into specified organizational unit.

This workflow action is used to move specified AD user into specified AD organizational unit or container.

Note: In order to specify AD credentials, add the "Set Active Directory Authentication (impersonate)" activity to the same workflow step first. Otherwise, AD will be accessed under context of the user who started the workflow or workflow author context in the case of impersonation step.

Parameters:

with this logon name - Logon name of the Active Directory user (e.g. DOMAIN\username).
specified - Distinguished name or canonical name of the Organizational Unit or Active Directory container (e.g. OU=Test,DC=company,DC=local or company.local/Users)

Examples:

Move Active Directory user COMPANY\user1 into company.local/Staff organizational unit.

1.20 Remove Member from Active Directory Group

Remove current user from members of specified group in Active Directory.

This workflow action will remove user or group from an Active Directory security group.

Note: In order to specify AD credentials, add the "Set Active Directory Authentication (impersonate)" activity to the same workflow step first. Otherwise, AD will be accessed under context of the user who started the workflow or workflow author context in the case of impersonation step.

Parameters:

current user - Logon name of the user or group: DOMAIN\user or user@domain.com. Leave the field empty to use current user.

specified - Logon name of the group: DOMAIN\group or group@domain.com.

Examples:

Remove COMPANY\jsmith from finance@company.com group in Active Directory

1.21 Set Active Directory Authentication

Set Active Directory authentication using user name this user and password this password, search in the current forest

This workflow action is used to specify AD credentials for all AD-related HarePoint activities (like "Remove Member from Active Directory Group" or "Delete Group from Active Directory") in the current workflow step. This activity must be placed above all other AD-related activities in the step. Otherwise, AD will be accessed under context of the user who started the workflow or workflow author context in the case of impersonation step.

Note: The password will be stored as a plain text in the workflow XOML file and may be accessed by the URL like *http://sharepoint/Workflows/workflowname/workflowname.xoml* or by SharePoint Designer.

Parameters:

this user - User name that will be used for Active Directory access (like DOMAIN\Account).

this password - Password that will be used for Active Directory access (see Note).

current forest - Specify the search scope to find the necessary user. Possible values:

- current
- current and trusted
- custom - in this case, additionally specify the forest DNS name

1.22 Set Local User's Password on Server

Set logon name account password on this server to this password .

This workflow action provides the capability to change the local user account password on specified server.

Important Note: *'Set Active Directory authentication'* action is required prior using this action to speciy the administrator account that will be used to create a local account

Parameters:

logon name - local account logon name

this server - NetBIOS name, DNS name or IP address.

Important Note: if you need to set a password for a user on a local machine, use *localhost* as a name!

this password - new password

Examples:

Set Active Directory authentication (impersonate) using name COMPANY\Administrator and password 111.

then Set Variable: logon name to test-n1

then Create new user on 192.168.2.16 server with Variable: logon name logon name, John Smith full name, Test user description, 123 password.

then Set Variable: logon name account password on 192.168.2.16 to 321.

1.23 Set User's Attribute in Active Directory

Set this attribute of current user in Active Directory to this value.

This workflow action is used to set attribute of specified user from Active Directory.

Note: In order to specify AD credentials, add the "Set Active Directory Authentication (impersonate)" activity to the same workflow step first. Otherwise, AD will be accessed under context of the user who started the workflow or workflow author context in the case of impersonation step.

See Also: "Set User's Multivalued Attribute in Active Directory" action.

Parameters:

this attribute - Name of attribute.
current user - User logon name (like DOMAIN\Account) or current user by default.
this value - Value for attribute.

1.24 Set User's Multivalued Attribute in Active Directory

Set multi-valued attribute this attribute of current user in Active Directory to this value.

This workflow action is used to set multi-valued attribute of the specified Active Directory user.

Note: This action uses array variable. Use "Create Array" or "Split Text" actions to create array variable.

Hint: This action can be used to set AD attribute to empty (null). To achieve this, create an empty array using "Create Array" action, then set the AD attribute to this array.

See Also: "Set User's Attribute in Active Directory" action.

Parameters:

this attribute - Name of attribute.
current user - User logon name (like DOMAIN\Account) or current user by default.
this value - Array variable with values.

1.25 Set User's Password in Active Directory

Set logon name account password in Active Directory to this password.

This workflow action is used to set password for specified Active Directory user.

Parameters:

logon name - Logon name of the user: DOMAIN\user or user@domain.com.
this password - String with new password.

Examples:

Set Active Directory authentication (impersonate) using user name Administrator and password adminpasswd.

then Set MAPILAB\testuser account password in Active Directory to newpassword.

1.26 Set User's Photo in Active Directory

Set current user's photo in Active Directory to photo at this url.

This workflow actions is used to set or remove photo in the Active Directory user's profile (thumbnailPicture attribute). JPG image format is required. The size of photo is limited to 100KB.

Parameters:

current user - Leave intact if you want to change photo of current user or select specific one.
this url - Full URL of photo. You can type "remove" or "delete" instead of URL to remove current user's photo.

2. Administration activities

2.1 Activate Feature

Activate this feature in feature collection at this url. Force: yes.

This workflow action is used to activate feature on SharePoint site or site collection.

Parameters:

this feature - Feature GUID or name, as it displayed at "Site Settings \ Manage site features" or at "Site Collection Administration \ Site collection features".
this url - Site URL.
Yes/No - Yes to force activation of the feature even if there are errors; otherwise, No.

2.2 Add Users to SharePoint Group

Add specified users to specified group at this site.

This action allows you to add one or more users to SharePoint group.

Parameters:

specified users - One or more logon names of users separated by semicolon.
specified group - Group name, as it displayed at People and Groups \ All Groups.
this - Full URL of a site.

2.3 Add Users to Site

Add specified users to this site

This workflow action is used to add local or Active Directory users to SharePoint site.

Parameters:

specified users - One or more logon user names separated by semicolon. For example, DOMAIN\account or MACHINE\account.
this site - Full URL of the site.

2.4 Change SharePoint Group Owner

Change specified group owner at current site url to current user account

This workflow action is used to change the owner of the SharePoint group.

Parameters:

specified group - specify the SharePoint group name
current site url - (optional) specify the address of the site where the group should be searched at.
current user - (optional) specify the name of the account or a group that should be set as a group owner

2.5 Create SharePoint Group

Create specified group at site url and make current user the owner, give this permission level, can view: Group Member, can edit: Group Owner.

This workflow action is used to create new SharePoint group. You can add members to the created group with "Add Users to SharePoint Group" action.

Parameters:

specified group - Group name.
site url - Full URL of the site.
current user - Logon name (like DOMAIN\account) of group owner. Current user by default.
this permission level - Site permissions for the new group. You can specify one or more permissions separated by semicolon, including custom permission levels as its listed at Site Settings \ Permission Levels.
can view: **Group Member** - Who can view the membership of the group. Available values: *Group Members, Everyone.*
can edit: **Group Owner** - Who can edit the membership of the group. Available values: *Group Owner, Group Members.*

2.6 Create SharePoint Site

Create new site with specified title at this url, using this template and this description, use unique permissions No. Store result in Variable: wasCreated .

This action allows you to create new site anywhere within the SharePoint environment.

You don't need to check in advance if the site exists, the action will check it automatically - see description for *Variable: wasCreated*

Parameters:

specified - Title of new site.
this url - Full URL for new site.
this template - Template for the new site. It might be name of site template, e.g. Team Site, Blank Site, Blog and etc. Either it might be ID of site template, e.g. STS#0, STS#1 and so on.
this description - New site description.

No - By default new site inherits permissions from parent Site or Site Collection or unique permissions can be created. If you select No, one set of user permissions is shared by both sites. Consequently, you cannot change user permissions on your new site unless you are an administrator of this parent site.

Variable: wasCreated - equals "True" if the site didn't exist and was created; equals "False" if the site already existed (no exception is thrown in this case).

2.7 Create Site Collection

Create new site collection with specified title at this url, using this template and specified description, compatibility level default. Site collection administrators: primary and secondary.

This action workflow action is used to create new site collection.

Note: May need to place this action into Impersonation Step. Otherwise, action will be executed under context of the user who started the workflow.

Parameters:

specified title - Title of new site collection.

this url - URL of new site collection.

this template - Template for the new site. It might be name of site template, e.g. Team Site, Blank Site, Blog and etc. Either it might be ID of site template, e.g. STS#0, STS#1 and so on.

specified description - Description of new site collection.

compatibility level **default** - set the compatibility level for site creation template.
For SharePoint 2013 only.
The list of available templates and their compatibility level can be obtained via **Get-SPWebTemplate**.

primary - Login name of administrator for this site collection. Security groups are not supported.

secondary - Optionally specify a secondary site collection administrator. Only one user login can be provided; security groups are not supported.

2.8 Deactivate Feature

Deactivate this feature in feature collection at this url. Force: yes.

This workflow action is used to deactivate feature on SharePoint site or site collection.

Parameters:

this feature - Feature GUID or name, as it displayed at "Site Settings \ Manage site features" or at "Site Collection Administration \ Site collection features".
this url - Full site URL.
Yes/No - Yes to force deactivation of the feature even if there are errors; otherwise, No.

2.9 Delete SharePoint Group

Delete specified group from this site url.

This workflow action is used to delete SharePoint groups from specified site.

Parameters:

specified - Name of group.
site url - Full URL of a site.

2.10 Delete SharePoint Site

Delete site at this url.

This action allows you to delete site anywhere within the SharePoint environment.

Parameters:

this url - Full URL of site to delete.

2.11 Delete Site Collection

Delete site collection at this url.

This workflow action is used to delete site collection.

Note: May need to place this action into Impersonation Step. Otherwise, action will be executed under context of the user who started the workflow.

Parameters:

this url - Full URL to site collection.

2.12 Get Multiple Site Users Property

Get (Custom) specified property for these users from user info list and store result in Variable:array

This workflow action gets the value of the specified property for all specified users and saves it to variable array

Parameters:

(Custom) - select the property from the drop-down list, or select (Custom) and specify the property display name in the next parameter
specified - specify the property display name. The list of default properties names are available here: http://technet.microsoft.com/en-us/library/hh147513(v=office.14).aspx
these users - users whose property should be received
user info list - select the source: User Information List, or Profile Service (not available on SharePoint Foundation). Note that workflow account should have permissions to be able to read from User Profile Service as follows: Central Administration -> Application Management -> Manage Service Applications -> User Profile Service -> Administrators
Variable:array - output array variable

Examples:

Get Manager property for VM1\User1; VM1\User2 from user info list and store result in Variable:array

2.13 Get Site User Property

Get (Custom) specified property for this user from user info list and store result in Variable: variable.

This workflow action is used to get specified property value from user's profile on the site.

For MOSS, you can get the list of all available properties in Central Administration \ Manage User Properties.

Parameters:

(Custom) - Select the property from the drop-down list, or select *Custom* and specify it in the next parameter
specified - Property display name.
this user - User whose property should be received
user info list - select the source: User Information List, or Profile Service (not available on SharePoint Foundation)
Note: Workflow account should have permissions to be able to read from User Profile Service as follows:
Central Administration -> Application Management -> Manage Service Applications -> User Profile Service -> Administrators
Variable: variable - Output string variable.

Examples:

Get About me property for Current Item:Created By from user info list and store result in Variable: am

*then Format Created field with * Create item at {0:dd-MM-yyyy hh:mm} and store in Variable: ds*

then Set Variable: ns to [%Variable: am%][%Variable: ds%]

then Set Variable: ns to About me property for Current Item: Created By

2.14 Is Role Assigned to User (HarePoint)

If role is assigned to this user

This condition is used to check is specified permission level (role) is assigned to user on the current site. This action obtains effective permission level by combining permissions granted directly to user with permissions granted to groups where user is a member.

Parameters:

role - User role (combination of permission levels).

this user - User logon name.

2.15 Is User a Member of a SharePoint Group (HarePoint)

If this user is a member of this SharePoint group

This condition allows you to check user membership in SharePoint group.

Parameters:

this user - User account name, like DOMAIN\logon.

this - SharePoint group name.

2.16 Remove User from SharePoint Group

Remove specified user from specified group at this site.

This workflow action is used to remove user from SharePoint group.

Parameters:

specified user - Logon of user to remove.

specified group - Group name, as it displayed at People and Groups \ All Groups.

this - Full URL of a site.

2.17 Remove User from Site

Remove specified user from this site.

This workflow action is used to remove user from SharePoint site and from all site groups.

Parameters:

specified - Logon name of user.

this - Full URL of a site.

2.18 Rename Site at Url

Rename site at this url to this title.

This workflow action is used to set new title for site specified by URL anywhere within the SharePoint environment.

Parameters:

this url - Full URL of site.

this title - New title.

2.19 Set Master Page

Set this master page as default for the current site.

This workflow action sets the master page for the current or any other site.

Parameters:

this masterpage - specify the name of the masterpage, or server-relative URL to it.

Examples:

- *v4.master*
- *default.master*
- */_catalogs/masterpage/v4.master*

the current site - (Optional) Specify the absolute URL of the site to set the masterpage to. Leave it *current site* to set the masterpage for the current site.

2.20 Set SharePoint List Anonymous Access

Set anonymous users can Add: **No**, Edit: **No**, Delete: **No**, View: **No** items at list at **list url**

This workflow action is used to set anonymous access for the list specified by URL. To use this action, anonymous access should be enabled for the web site: http://office.microsoft.com/en-au/windows-sharepoint-services-help/enable-anonymous-access-HA010113018.aspx

Parameters:

Add: **No** - Anonymous user can add items. Available values: *Yes, No*.
Edit: **No** - Anonymous user can edit items. Available values: *Yes, No*.
Delete: **No -** Anonymous user can delete items. Available values: *Yes, No*.
View: **No** - Anonymous user can view items. Available values: *Yes, No*.
list url - Full URL of target list.

2.21 Set SharePoint Site Anonymous Access

Set anonymous users can access: Nothing at site url.

This workflow action is used to set anonymous access for the site specified by URL. To use this action, anonymous access should be enabled for the web site: http://office.microsoft.com/en-au/windows-sharepoint-services-help/enable-anonymous-access-HA010113018.aspx

Parameters:

Nothing - What anonymous users can access at target web site. Available values: *Nothing, Lists and Libraries, All Web Site*.
site url - Full URL of target web site.

2.22 Set SharePoint Site Navigation Options

Set options for site at this url, show on parent's Quick Launch: yes , show on parent's top link bar: yes , inherit parent's top link bar: yes .

This workflow action sets the navigation options for the SharePoint site.

Parameters:

site at this url - specify the site URL to set the options for.
show on parent's Quick Launch: yes - specify if you need to display or hide this site from Quick Launch of the parent site.
Available values: *yes, no, don't change* (to leave as it currently is)

show on parent's top link bar: yes - specify if you need to display or hide this site from Top Link Bar of the parent site.

Available values: *yes, no, don't change* (to leave as it currently is)

inherit parent's top link bar: yes - choose to display the Top Link Bar from the parent site, or display the own Top Link Bar for this site.

Available values: *yes, no*

2.23 Set Site User Property

Set value to specified property for this user

This workflow action is used to set specified property value in user's profile on the site.

For MOSS, you can get the list of all available properties in Central Administration \ Manage User Properties.

See Also: "Set User's Multivalued Attribute in Active Directory" action.

Parameters:

value - Text, expression, item field or variable.

specified - Property display name.

this user - User, in whose profile property value is setting.

Examples:

The workflow below allows to user automatically change his picture in profile by uploading JPG file to documents library.

If Current Item:File Type equals jpg

then Set [%Current Item:URL Path%] to Picture property for Current Item:Created By

2.24 Set User's Photo in SharePoint

Set current user's photo in SharePoint profile to photo at this url. Copy file to User Information List

This workflow actions is used to set or remove photo in the SharePoint user's profile. JPG image format is required. The size of photo is limited to 1MB.

Parameters:

current user - Leave intact if you want to change photo of current user or select specific one.
this url - Full URL of photo. You can type "remove" or "delete" instead of URL to remove current user's photo.
Copy file - select to copy/do not copy this photo to User Information List

3. Arrays activities

3.1 Add Array Item

Add item to the end of the array variable

This workflow action is used to add a new item to the end of array or to copy source array specified in the first parameter to the end of the target array specified in the second parameter.

Parameters:

item - New item to add or source array to copy.
array variable - Target array variable.

3.2 Clear Array

Remove all items from the array variable.

This workflow action is used to remove all items from specified array variable.

Parameters:

array variable - Array variable.

3.3 Create Array

Create String array with initial items and store in Variable: ArrayVariable

This workflow action is used to create array variable.

Parameters:

String - Type of the array. Available types are: *String*, *Date/Time* - date and time values, *Integer* - positive or negative integer values, *Number* - floating point numbers.

initial - One or more items for array separated by semicolon.

Variable: ArrayVariable - Output array variable.

Examples:

Create String array with John;Mark;Alex items and store in Variable: array1

then Create String array with Jane;Kate;Olga items and store in Variable: array2

then Insert Variable: array2 into Variable: array1 at 1 index

then Get index of Mark item in Variable: array1 and store in Variable: index

then Log Variable: index to the workflow history list

Result in workflow history list: 4

3.4 Get Array Item

Get item at 0 index from array variable and store in variable.

This workflow action is used to get an item at specified index of array.

Non-negative values of index are zero-based and means 'from beginning of array', so 1 means 'second item of array'.

Negative value of index means 'from the end of array', so -1 means 'last item of array'.

Parameters:

0 - Index of array item.

array variable - Array variable.

variable - Output variable to store result.

3.5 Get Array Items Count

Get the number of items in the array variable and store in Variable: variable1

This action allows you to get the number of array items.

Parameters:

array variable - Array variable.
Variable: variable1 - Output integer variable.

Examples:

Get list of the Active Directory groups where user current user is member. Store list of these groups in Variable: GroupsArray, success of searching in Variable: UserFound.

then Get the number of items in the Variable: GroupsArray and store in Variable: groupsCount.

then Log [%Workflow Context: Current User%] is the member of [%Variable: groupsCount%] groups to the workflow history list

3.6 Get Index of Array Item

Get index of item item in the array variable and store in Variable: variable1

This workflow action is used to get zero-based index of specified item in array variable.

Result -1 means that item is not found.

Parameters:

item - Item to get index of.
array variable - Array variable.
Variable: variable1 - Output integer variable to store result. -1 means that item is not found.

Examples:

Create String array with John;Mark;Alex items and store in Variable: array1

then Create String array with Jane;Kate;Olga items and store in Variable: array2

then Insert Variable: array2 into Variable: array1 at 1 index

then Get index of Mark item in Variable: array1 and store in Variable: index

then Log Variable: index to the workflow history list

Result in workflow history list: 4

3.7 Insert Array Item

Insert array item into array variable at 0 index.

This workflow action is used to insert new item to specified position of array variable. Or to insert source array specified in the first parameter to the specified position of target array specified in second parameter.

Parameters:

item - New item or source array to insert.

array variable - Target array variable.

0 - Zero-based array index.

Examples:

Create String array with John;Mark;Alex items and store in Variable: array1

then Create String array with Jane;Kate;Olga items and store in Variable: array2

then Insert Variable: array2 into Variable: array1 at 1 index

then Get index of Mark item in Variable: array1 and store in Variable: index

then Log Variable: index to the workflow history list

Result in workflow history list: 4

3.8 Remove Array Item

Remove item from the array variable

This workflow action is used to delete first occurrence of specified item from array variable.

Parameters:

item - Item to remove from array.

array variable - Array variable.

3.9 Remove Array Item at

Remove the item at 0 index from array variable

This workflow action is used to remove item at specified zero-based index from array.

Parameters:

0 - Zero-based index of array item.

array variable - Array variable.

3.10 Remove Duplicates from Array

Remove duplicates from the array variable

This workflow action is used to delete duplicated values in array.

Parameters:

array variable - Array variable.

4. Date/Time activities

4.1 Converts Date from Local Time to UTC

Converts specified date from Local Time to UTC and store result in Variable: ConvertedToUTCDate.

This workflow action converts local date/time variable to Coordinated Universal Time (UTC) date/time. The UTC is equal to the local time minus the UTC offset. The conversion also takes into account the daylight saving time rule.

Parameters:

specified date - Input date or Date/Time variable.

Variable: ConvertedToUTCDate - Output Date/Time variable.

4.2 Converts Date from UTC to Local Time

Converts specified date from UTC to Local Time and store result in Variable: ConvertedToLocalTimeDate

This workflow action is used to convert Coordinated Universal Time (UTC) to local time. The local time is equal to the UTC time plus the UTC offset. The conversion also takes into account the daylight saving time rule.

Parameters:

specified date - Input UTC date or date/time variable
Variable: ConvertedToLocalTimeDate - Output local time/date variable

4.3 Converts Date to Text

Converts specified date to text with specified format and store result in Variable: TextDate

This workflow action is used to convert date/time variable to string using specified format.

Detailed description of format string syntax is contained in following articles:

- Standard DateTime Format Strings - http://msdn.microsoft.com/en-us/library/aa720651(v=VS.71).aspx
- Custom DateTime Format Strings - http://msdn.microsoft.com/en-us/library/aa719856(v=VS.71).aspx
- DateTime Formatting Sample - http://msdn.microsoft.com/en-us/library/aa326720(v=VS.71).aspx

In the case of the wrong format string, the date will be converted with default format and warning message will be stored into workflow log file.

Parameters:

specified date - Input date/time variable.
specified format - Format string.
Variable: TextDate - Output string variable.

Examples:

Set DateTime constant Now and store result in Variable: d

then Convert Variable: d to text with MMMM dd yyyy HH:mm:ss.fffffff and store result in Variable: strDate

then Get Variable: d day property and store result in Variable: day

then Get Variable: d day of week property and store result in Variable: dow

then Get Variable: d day of year property and store result in Variable: doy

then Get Variable: d hour property and store result in Variable: h

then Get Variable: d millisecond property and store result in Variable: ms

then Get Variable: d minute property and store result in Variable: min

then Get Variable: d month property and store result in Variable: m

then Get Variable: d second property and store result in Variable: sec

then Get Variable: d year property and store result in Variable: y

then Log [%Variable: strDate%] : [%Variable: day%], [%Variable: dow%], [%Variable: doy%], [%Variable: h%], [%Variable: ms%], [%Variable: min%], [%Variable: m%], [%Variable: sec%], [%Variable: y%] to the workflow history list

Result in workflow history file:

April 25 2011 14:16:26.1626432 : 25, 2, 115, 14, 162, 16, 4, 26, 2011

4.4 Evaluate Due Date

Evaluate due date from this time for this many hours, this many minutes and store result in Variable: variable.

This workflow action is used to calculate due date for the specific period excluding non-working hours.

Parameters:

this time - Starting date/time.
this many hours - Duration in hours.
this many minutes - Duration in minutes.
Variable: variable - Output date/time variable.

4.5 Get Date Property

Get this date day property and store result in Variable: DateProperty.

This workflow action is used to get one from numeric values (like day of week or year) of date/time variable. See examples section for details.

Parameters:

this date - Input date/time variable.
day - What number from the date to store: *day, day of week, day of year, hour, millisecond, minute, month, second, year*. See examples section for details.
Variable: DateProperty - Output integer variable.

Examples:

Set DateTime constant Now and store result in Variable: d

then Convert Variable: d to text with MMMM dd yyyy HH:MM:ss.fffffff and store result in Variable: strDate

then Get Variable: d day property and store result in Variable: day

then Get Variable: d day of week property and store result in Variable: dow

then Get Variable: d day of year property and store result in Variable: doy

then Get Variable: d hour property and store result in Variable: h

then Get Variable: d millisecond property and store result in Variable: ms

then Get Variable: d minute property and store result in Variable: min

then Get Variable: d month property and store result in Variable: m

then Get Variable: d second property and store result in Variable: sec

then Get Variable: d year property and store result in Variable: y

then Log [%Variable: strDate%] : [%Variable: day%], [%Variable: dow%], [%Variable: doy%], [%Variable: h%], [%Variable: ms%], [%Variable: min%], [%Variable: m%], [%Variable: sec%], [%Variable: y%] to the workflow history list

Result in workflow history list:

April 25 2011 14:04:26.1626432 : 25, 2, 115, 14, 162, 16, 4, 26, 2011

4.6 Get Interval Between Dates

Get minutes between this date and this date, store interval in Variable:interval

This workflow action calculates the interval between two specified dates in specified units and saves the result to the variable.

Parameters:

get **minutes** - select the unit for the time interval. Possible values: *minutes, hours, days, weeks, months, years*.

between **this date** - the starting date

and **this date** - the ending date

Variable:interval - the Number variable where result will be stored

4.7 Set DateTime Constant

Set DateTime constant Now and store result in Variable: DateConstant.

This workflow action is used to initialize date/time variables. See parameters and examples sections for details.

Parameters:

Now - One from following date/time values: *Now, UtcNow, Today, Yesterday, Tomorow, Beginning of this Week, End of this Week, Beginning of next Week, End of next Week, Beginning of this Month, End of this Month, Beginning of next Month, End of next Month, Beginning of this Year, End of this Year, Beginning of next Year, End of next Year*. See examples section for details.

Variable: DateConstant - Output date/time variable.

Examples:

Set DateTime constant Now and store result in Variable: vNow

then Set DateTime constant Today and store result in Variable: vToday

then Set DateTime constant End of this Week and store result in Variable: vEndOfWeek

then Set DateTime constant Beginning of next Year and store result in Variable: vBeginningNextYear

*then Log [%Variable: vNow%] * [%Variable: vToday%] * [%Variable: vEndOfWeek%] * [%Variable: vBeginningNextYear%] to the workflow history list*

Result in workflow history list:

4/25/2011 2:56:12 PM * 4/25/2011 12:00:00 AM * 4/30/2011 12:00:00 AM * 1/1/2012 12:00:00 AM

5. Development activities

5.1 Execute Custom Code

Execute this code in C# with these referenced assemblies and store result in variable.

This workflow action is used to execute custom C# or VB.NET code within workflow.

In the code, the object named Context with IExecuteCustomCodeLink interface is available. With this object, you can easily get access to workflow context, output Result variable and to any other workflow variable, activity execution context, and workflow history list (using LogComment function).

```
public interface IExecuteCustomCodeLink
{
    WorkflowContext WorkflowContext { get; }
    object Result { get; set; }
    object GetWorkflowVariable(string variableName);
    void SetWorkflowVariable(string variableName, object value);
    void LogComment(string comment);
    ActivityExecutionContext ExecutionContext { get; }
}
```

See examples section below.

Note: This workflow action is not safe. If you need to disable it due to security reasons, see article "How to disable certain HarePoint activities" in Section III of this Manual.

Parameters:

this code - C# or VB.NET code, see examples below. Use *using* C# keyword or *Imports* VB.NET statement to get access to additional namespaces. The following namespaces are already available in the code: System, Microsoft.SharePoint, System.Workflow.ComponentModel.

C# - Choose of two available languages: C# or VB.NET.

these referenced assemblies - Optional multi-line string with assemblies. Each line should contain a single file name. References to the following assemblies already exist even when this parameter is empty: System.dll, System.Data.dll, System.Xml.dll, System.Workflow.Activities.dll, System.Workflow.ComponentModel.dll, Microsoft.Sharepoint.WorkflowActions.dll, Microsoft.SharePoint.dll, HarePoint.Activities.dll.

It is possible to specify assemblies from Global Assembly Cache. It is recommended to use a full name:

Multiplier35, Version=1.0.0.0, Culture=neutral, PublicKeyToken=83a9124cb39a6085

but shorter names are also allowed:

Multiplier35 or Multiplier35.dll

variable - Output variable. This variable is accessible from code as *Context.Result*.

Examples:

Set DateTime constant Now and store result in Variable: MyDate

the Execute int DateShift = 10; DateTime odt = ... in C# with these referenced assemblies and store result in Variable: DateShift

then Log The date is moved for [%Variable: DateShift%] years, to [%Variable: MyDate%] to the workflow history list

This sample moves the date in MyDate variable for ten years from the code. The complete code for Execute Custom Code action is the following:

```
int DateShift = 10;
DateTime odt = (DateTime)Context.GetWorkflowVariable("MyDate");
DateTime ndt = odt.AddYears(DateShift);
Context.SetWorkflowVariable("MyDate", ndt);
Context.Result = DateShift;

Context.LogComment("Date is moved from custom code");
```

Result in workflow history list:

Date is moved from custom code

The date is moved for 10 years, to 6/30/2021 5:49:54 PM

5.2 Execute PowerShell Script

Execute this PowerShell script and store result in Variable: results. Run as application pool account.

This workflow action is used to execute custom PowerShell script within a workflow.

In the code, the object named $Context is available. With this object, you can easily get access to workflow context, output Result variable and to any other workflow variable, activity execution context, and workflow history list (using Write-Host).

Additionally, two variables with user credentials available in the code: $credential - NetworkCredential object, $pscredential - PSCredential object.

See examples section below.

Note: This workflow action is not safe. If you need to disable it due to security reasons, see article "How to disable certain HarePoint activities" in Section III of this Manual.

Parameters:

this PowerShell script - PowerShell script, see example below.

Variable: results - Output variable. This variable is accessible from code as $Context.Result.

Run as application pool account - specify the account that will be used to run the PowerShell script. You can use **Encrypt Credentials** tool and 'Set Encrypted Passwords' action to provide the credentials for alternative account in secure way; this tool is installed as a Client Component using the HarePoint Workflow Extensions installer.

Examples:

Set encrypted passwords with sfDdv1yb+3... string

Execute $x = $Context.GetWorkflowVariable("v... and store result in Variable: results. Run as encrypted password account.

The complete code for Execute PowerShell Script action is the following:

```
$x = $Context.GetWorkflowVariable("v1")
$Context.LogComment("test comment")
Write-Host "test"
```

```
$Context.Result = "result string"
$Context.SetWorkflowVariable("v1", "sample_value")
Write-Host $Context.WorkflowContext.Web.Title
Get-ExecutionPolicy -Scope Process
Get-ExecutionPolicy -Scope CurrentUser
Get-ExecutionPolicy -Scope LocalMachine
Get-ExecutionPolicy
```

5.3 Execute SQL Command

Execute this SQL command using specified connection string for MS SQL database and store result in Variable: variable, number of records in Variable: recordsAffected.

This workflow action is used to execute SQL command on Microsoft SQL Server, Oracle, ODBC or OLE DB data sources.

For SQL language reference, see your DMBS documentation.

Parameters:

this SQL command - String, expression or variable with vaild SQL command.
specified connection string - Database connection string. See your DMBS documentation for connection string format.
MS SQL - Type of database: MS SQL, ODBC, OLE DB or Oracle.
Variable: variable - Output array variable for retrieved records. If only one record was retrieved by SELECT command (value of *recordsAffected* is 1) then this array contains the record. If more than one record was retrieved by SELECT command (value of *recordsAffected* is 2 or greater) then this array contains values of the first column of recordset.
Variable: recordsAffected - Output integer variable with the number of records affected by command execution.

Examples:

Execute INSERT INTO [Sheet1$] (ModifiedAt, FileName, UserName) VALUES ('[%Current Item:Modified%]', '[%Current Item:Name%]', '[%Current Item:ModifiedBy%]'); using DSN=FileDb for ODBC database and store result in Variable: variable2, number of records in Variable: affectedRecords

then Execute SELECT TOP 1 ModifiedAt, UserName FROM [Sheet1$] WHERE FileName='[%Current Item:Name%]' ORDER BY ModifiedAt using DSN=FileDb for ODBC database and store result in Variable: variable3, number of records in Variable: affectedRows

then Get item at 0 index from Variable: variable3 and store in Variable: FirstMod

If Variable: FirstMod not equals Current Item:Modified

Get item at 1 index from Variable: variable3 and store in Variable: FirstModificationBy

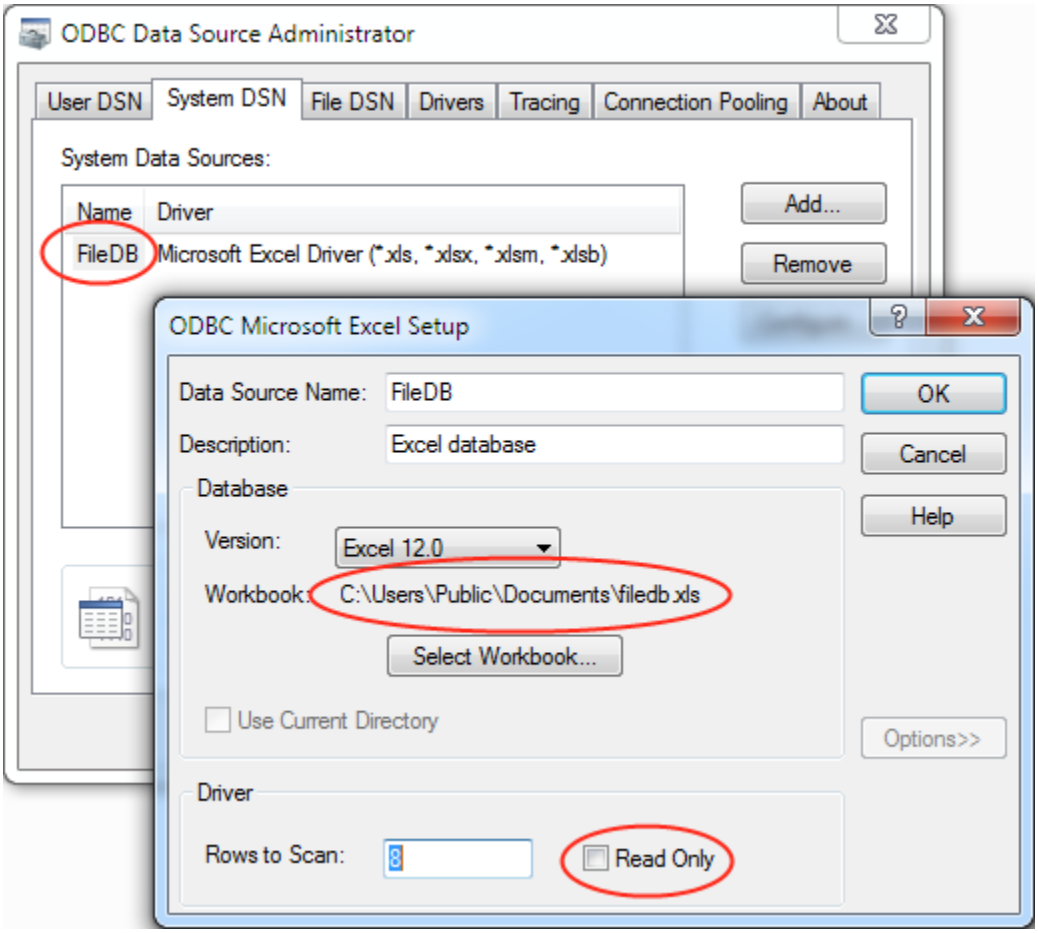
then Log This document was first time created by [%Variable: FirstModificationBy%] at [%Variable: FirstMod%] to the workflow history list

This sample uses a simple Excel file with the structure shown below as the database.

	A	B	C
1	**ModifiedAt**	**FileName**	**UserName**
2	6/30/2011 3:53:20 PM	2011-06-04 15-48-35 - Sasha - IMG_7591	GORLACH7\gorlach
3	7/1/2011 12:07:04 PM	2011-06-04 15-47-33 - Sasha - IMG_7588	GORLACH7\gorlach
4	7/1/2011 12:09:04 PM	2011-06-04 15-47-57 - Sasha - IMG_7590	GORLACH7\gorlach
5	7/1/2011 12:09:51 PM	2011-06-04 15-47-54 - Sasha - IMG_7589	GORLACH7\gorlach
6	7/1/2011 12:11:57 PM	2011-06-04 15-48-35 - Sasha - IMG_7591	GORLACH7\gorlach
7	7/1/2011 12:13:50 PM	2011-06-04 15-48-37 - Sasha - IMG_7592	GORLACH7\gorlach
8	7/1/2011 12:14:41 PM	2011-06-04 15-47-21 - Sasha - IMG_7586	GORLACH7\gorlach
9	7/1/2011 12:16:06 PM	2011-06-05 15-53-43 - Sasha - IMG_7600	GORLACH7\gorlach
10	7/1/2011 12:16:16 PM	2011-06-04 15-47-24 - Sasha - IMG_7587	GORLACH7\gorlach
11	7/1/2011 12:17:26 PM	2011-06-04 15-47-33 - Sasha - IMG_7588	GORLACH7\gorlach
12			

Sheet1 Sheet2 Sheet3

To run this sample, you should also create System DSN with the FileDB name for the file at Administrative Tools \ Data Sources (ODBC).



Configure this workflow to start automatically when document is added or changed in document library and it will log all activity with the documents to an Excel file.

Result in workflow history list:

This document was first time created by GORLACH7\gorlach at 7/1/2011 12:07:04 PM

5.4 Execute SQL Command into List

Execute this SQL command using specified connection string for specified database and add results into list at this url using this column mapping, store number of records in Variable: recordsAffected.

This workflow action is used to execute SQL command on Microsoft SQL Server, Oracle, ODBC or OLE DB data sources and store results in the specified SharePoint list. Each row from the results represents new list item. You can specify alternate column mapping in the fifth parameter.

For example, you have a list with two columns named *Title* and *Due Date*. And you have following SQL query: *SELECT Name, Date from Table*. Then you can set column mapping to the following:

Name=Title

Date=Due Date

Parameters:

this SQL command - String, expression or variable with vaild SQL command.
specified connection string - Database connection string. See your DMBS documentation for connection string format.
specified database - Type of database: *MS SQL*, *ODBC*, *OLE DB* or *Oracle*.
this url - Full URL of the list where items should be added.
this column mapping - This parameter may be blank. Multiline string with alternate column mapping, each on new line, in the following format:
DB_Column_Name1=SP_Column_Name1
DB_Column_Name2=SP_Column_Name2
Variable: recordsAffected - Output integer variable with the number of records affected by command execution.

5.5 Query XML

Query this XML using XPath query this query and store resulting text in Variable: variable

This workflow action is used to query XML document using XPath query language.

See Also:

- XPath Language Reference: http://www.w3.org/TR/xpath/
- XPath Online Tutorial: http://www.w3schools.com/xpath/default.asp
- Online XPath Query Tool: http://www.bit-101.com/xpath/

Parameters:

this XML - Input string, expression or variable with XML document.

this query - XPath query string. See examples section for details.
text - One from three available options: *text*, *inner XML* or *outer XML*.
Variable: variable - Output string variable.

Examples:

Send HTTP request to http://www.ecb.int/stats/eurofxref/eurofxref-daily.xml using GET method with this headers. Store response in Variable: CurrencyXml and status code in Variable: variable

then Query [%Variable: CurrencyXml%] using XPath query //Cube[@currency='USD']/@rate and store resulting text in Variable: EuroUsd

then Log European Central Bank EUR to USD conversion rate is: [%Variable: EuroUsd%] to the workflow history list

This sample takes XML document with the currency conversion rates from European Central Bank, and gets actual EUR/USD conversion rate.

Result in the workflow history list:

The request was sent successfully: http://www.ecb.int/stats/eurofxref/eurofxref-daily.xml

European Central Bank EUR to USD conversion rate is: 1.4453

Example of XML document of European Central Bank:

<gesmes:Envelope xmlns:gesmes="http://www.gesmes.org/xml/2002-08-01" xmlns="http://www.ecb.int/vocabulary/2002-08-01/eurofxref">

<gesmes:subject>Reference rates</gesmes:subject>

<gesmes:Sender>

<gesmes:name>European Central Bank</gesmes:name>

</gesmes:Sender>

<Cube>

<Cube time="2011-06-30">

<Cube currency="USD" rate="1.4453"/>

<Cube currency="JPY" rate="116.25"/>

<Cube currency="BGN" rate="1.9558"/>

<Cube currency="CZK" rate="24.345"/>

<Cube currency="DKK" rate="7.4587"/>

<Cube currency="GBP" rate="0.90255"/>

</Cube>

</Cube>

</gesmes:Envelope>

5.6 Transform XML

Transform this XML using XSLT this transform and store result in Variable: variable

This workflow action is used for XML transformations using XSLT template.

See Also:

- XSL Transformations Version 1.0: http://www.w3.org/TR/xslt
- Online XSLT Tutorial: http://www.w3schools.com/xsl/default.asp
- Online XSLT Design Tool: http://www.w3schools.com/xsl/tryxslt.asp xmlfile=cdcatalog&xsltfile=cdcatalog

Parameters:

this XML - Input string, expression or variable with XML document.
this transform - Input string, expression or variable with XSLT template.
Variable: variable - Output string variable.

Examples:

Send HTTP request to http://www.ecb.int/stats/eurofxref/eurofxref-daily.xml using GET method with this headers. Store response in Variable: Xml and status code in Variable: Code

then Transform [%Variable: Xml%] using XSLT <xsl:stylesheet version="1.0" xmlns:xsl=... and store result in Variable: Html

then Create document from Variable: Html in http://gorlach7/Shared%20Documents/Rates.Html. Overwrite existing document.

This sample takes XML document with currency conversion rates from European Central Bank and generates HTML document with currency rates in document library.

For a sample of XML document, please see the examples section of 'Query XML' workflow action overview.

Full text of XSLT template for this sample:

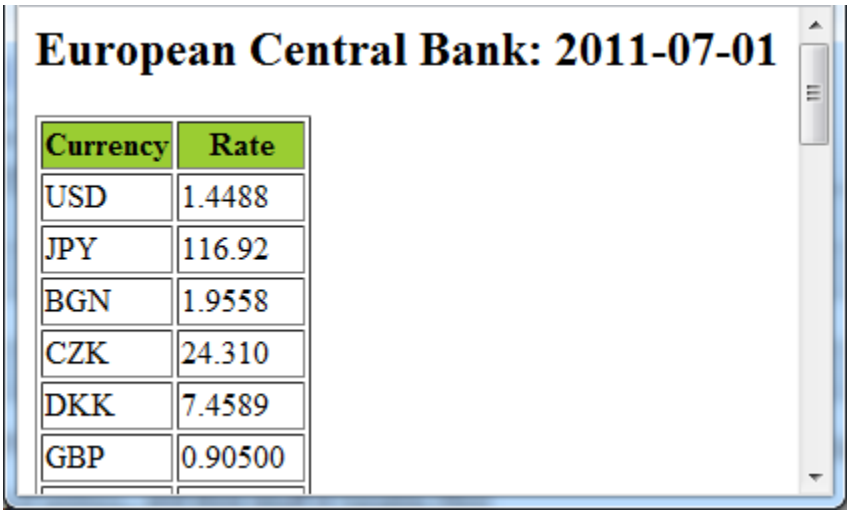
```
<xsl:stylesheet version="1.0"

xmlns:xsl="http://www.w3.org/1999/XSL/Transform"

xmlns:gesmes="http://www.gesmes.org/xml/2002-08-01"

xmlns:cubes="http://www.ecb.int/vocabulary/2002-08-01/eurofxref">

<xsl:template match="/">

 <html>

 <body>

 <h2><xsl:value-of select="//gesmes:name"/>: <xsl:value-of select="//cubes:Cube/@time" /></h2>

  <table border="1">

   <tr bgcolor="#9acd32">

    <th>Currency</th>

    <th>Rate</th>

   </tr>

   <xsl:for-each select="//cubes:Cube/cubes:Cube/cubes:Cube">

   <tr>

    <td><xsl:value-of select="@currency"/></td>

    <td><xsl:value-of select="@rate"/></td>

   </tr>

   </xsl:for-each>

  </table>

 </body>
```

</html>

</xsl:template>

</xsl:stylesheet>

Example of output HTML document:



European Central Bank: 2011-07-01

Currency	Rate
USD	1.4488
JPY	116.92
BGN	1.9558
CZK	24.310
DKK	7.4589
GBP	0.90500

5.7 Update XML

Update this XML, update node value using XPath this query with this value, store result in Variable: variable

This workflow action is used to update node value or append new child node to XML document using XPath query.

See Also:

- XPath Language Reference: http://www.w3.org/TR/xpath/
- XPath Online Tutorial: http://www.w3schools.com/xpath/default.asp
- Online XPath Query Tool: http://www.bit-101.com/xpath/

Parameters:

this XML - Input text, expression or variable with XML document.
update node value - Two options are available: *update node value* or *append child node*.
this query - Input text, expression or variable with XPath query.

this value - New node value or new child node XML.

Variable: variable - Output string variable for updated XML document.

Examples:

1. Update node sample

Send HTTP request to http://www.ecb.int/stats/eurofxref/eurofxref-daily.xml using GET method with this headers. Store response in Variable: Xml and status code in Variable: Code

then Query [%Variable: Xml%] using XPath query //Cube[@currency='USD']/@rate and store resulting text in Variable: EuroUsd

then Set Variable: EuroUsdNumber to Variable: EuroUsd

then Calculate Variable: EuroUsdNumber multiply by 1,1 (Output to Variable: EuroUsdCorrected)

then Update [%Variable: Xml%], update node value using XPath //Cube[@currency='USD']/@rate with [%Variable: EuroUsdCorrected%], store result in Variable: NewXml

then Transform [%Variable: NewXml%] using XSLT <xsl:stylesheet version="1.0" xmlns:xsl=... and store result in Variable: Html

then Create document from Variable: Html in http://gorlach7/Shared%20Documents/Rates.Html.Overwrite existing document.

This sample takes XML document with currency conversion rates from European Central Bank, increases EUR/USD conversion rate by 10%, and generates HTML document with currency rates in document library.

Note that type of EuroUsdNumber variable is Number. For the full text of XSLT template and example of output HTML see examples section of 'Transform XML' activity overview. For the sample of source XML document see examples section of 'Query XML' activity overview.

2. Append child node sample

Send HTTP request to http://www.ecb.int/stats/eurofxref/eurofxref-daily.xml using GET method with this headers. Store response in Variable: Xml and status code in Variable: Code

then Update [%Variable: Xml%], append child node value using XPath //Cube/Cube[@currency='USD'] with <Cube currency="EUR" rate="1.0000"/>, store result in Variable: NewXml

then Transform [%Variable: NewXml%] using XSLT <xsl:stylesheet version="1.0" xmlns:xsl=... and store result in Variable: Html

then Create document from Variable: Html in http://gorlach7/Shared%20Documents/Rates.Html.Overwrite existing document.

This sample takes XML document with currency conversion rates from European Central Bank, add new node with EUR/EUR rate 1.0000, and generates HTML document with currency rates in document library.

For the full text of the XSLT template and example of output HTML, see examples section of the 'Transform XML' activity overview. For a sample of source XML document, please see examples section of the 'Query XML' activity overview.

6. Document Conversion activities

6.1 Convert Document

Convert this document to PDF format and store job ID in Variable: ConversionJobId. Overwrite existing document.

This workflow action is used to convert specified document to different format and store output file in the same document library.

This action process only the files of supported formats, files of other formats will be skipped. List of supported input formats:

- Open XML Document (.docx, .docm, .dotx, .dotm)
- Word 97-2003 Document (.doc, .dot)
- Rich Text Format (.rtf)
- Web Page (.htm, .html, .mht, .mhtml)
- Word 2003 XML Document (.xml)

List of supported output formats includes all input formats and two additional:

- Portable Document Format (.pdf)
- XML Paper Specification (.xps)

Action returns empty string in ConversionJobId output variable is no files were found to convert.

Note: This action require "Word Automation Services" running on server. To check this service, open Central Administration, go to System Settings and open "Manage services on server" page. Unfortunately, this service is not available on SharePoint Foundation.

See Also: "Wait Conversion Job" and "Get Conversion Job Status" actions.

Parameters:

this - Document to convert, e.g. Current Item.
PDF - Choose one from available formats: *Word Document (docx), Word 97-2003 Document (doc), PDF, RTF, XPS*.
Variable: ConversionJobId - Output string variable with ID of conversion job. Action returns empty string if no files were found for conversion.
Overwrite - If the folder already contains file with the same name, then you can apply one from two options. *Overwrite* - replace file by new one. *Do not overwrite* - skip this file.

Examples:

See example at "Wait Conversion Job" action description.

6.2 Convert Folder at Url

Convert documents in folder at this url to PDF format and store documents to folder at specified url, store job ID in Variable: ConversionJobId. Overwrite existing document.

This workflow action is used to convert all documents from document library specified by URL. Output files maybe stored in the same or at another document library.

This action process only the files of supported formats, files of other formats will be skipped. List of supported input formats:

- Open XML Document (.docx, .docm, .dotx, .dotm)
- Word 97-2003 Document (.doc, .dot)
- Rich Text Format (.rtf)
- Web Page (.htm, .html, .mht, .mhtml)
- Word 2003 XML Document (.xml)

List of supported output formats includes all input formats and two additional:

- Portable Document Format (.pdf)
- XML Paper Specification (.xps)

Action returns empty string in ConversionJobId output variable is no files were found to convert.

Note: This action require "Word Automation Services" running on server. To check this service, open Central Administration, go to System Settings and open "Manage services on server" page. Unfortunately, this service is not available on SharePoint Foundation.

See Also: "Wait Conversion Job" and "Get Conversion Job Status" actions.

Parameters:

this url - Full URL of source document library or URL of a folder of document library.
PDF - Choose one from available formats: *Word Document (docx), Word 97-2003 Document (doc), PDF, RTF, XPS*.
specified url - Full URL of target document library or URL of a folder of document library.
Variable: ConversionJobId - Output string variable with ID of conversion job. Action returns empty string if no files were found for conversion.
Overwrite - If the folder already contains file with the same name, then you can apply one from two options. *Overwrite* - replace file by new one. *Do not overwrite* - skip this file.

6.3 Convert InfoPath Form and Save to Document Library

Convert InfoPath form at list item url to PDF and save to document library at library url

This workflow action converts the filled InfoPath form at specified Form Library item to PDF and saves to a specifed Document Library.

Important note: This action requires a special setup before using it. Please refer to '*InfoPath to PDF Service Installation Guide*' and '*InfoPath to PDF How to use*' guide for detailed information. These guides can be found in **Docs** folder in the distributive package.

Parameters:

list item url - specify a list item url that should be processed
PDF - select the target format, currently PDF is the only available option

library url - specify the URL for the library where coverted InfoPath forms will be created

Examples:

Convert InfoPath form at Current Item:Encoded Absolute URL to PDF and save to document library at http://labsp13/Converted%20Documents

6.4 Convert Library

Convert documents in this library to PDF format and store converted documents to specified library, store job ID in Variable: ConversionJobId. Overwrite existing document.

This workflow action is used to convert all documents from the document library. Output files maybe stored in the same or at another document library.

This action process only the files of supported formats, files of other formats will be skipped. List of supported input formats:

- Open XML Document (.docx, .docm, .dotx, .dotm)
- Word 97-2003 Document (.doc, .dot)
- Rich Text Format (.rtf)
- Web Page (.htm, .html, .mht, .mhtml)
- Word 2003 XML Document (.xml)

List of supported output formats includes all input formats and two additional:

- Portable Document Format (.pdf)
- XML Paper Specification (.xps)

Action returns empty string in ConversionJobId output variable is no files were found to convert.

Note: This action require "Word Automation Services" running on server. To check this service, open Central Administration, go to System Settings and open "Manage services on server" page. Unfortunately, this service is not available on SharePoint Foundation.

See Also: "Wait Conversion Job" and "Get Conversion Job Status" actions.

Parameters:

this library - Source document library on current site.

PDF - Choose one from available formats: *Word Document (docx), Word 97-2003 Document (doc), PDF, RTF, XPS*.

specified library - Target document library on current site.

Variable: ConversionJobId - Output string variable with ID of conversion job. Action returns empty string if no files were found for conversion.

Overwrite - If the folder already contains file with the same name, then you can apply one from two options. *Overwrite* - replace file by new one. *Do not overwrite* - skip this file.

6.5 Get Conversion Job Status

Get conversion job with this job ID status and store in Variable: JobStatus; store links to converted documents in Variable: DocumentsArray.

This workflow action is used to get a string with status of conversion job, one from three values is possible: *Completed*, *Failed* or *Running*. For completed job, it also returns array with URLs of new documents.

See Also: "Wait Conversion Job" and "Get Array Items Count" actions.

Parameters:

with this job ID - ID of conversion job returned by "Convert Document", "Convert Folder at Url" or "Convert Library" actions.

Variable: JobStatus - Output string variable for the job status, contains one from three possible values: *Completed*, *Failed* or *Running*.

Variable: DocumentsArray - Output array variable for the full URLs of new files. Use "Join Text" or "Get Array Item" actions to get string values.

Examples:

See example at "Wait Conversion Job" action description.

6.6 Wait Conversion Job

Wait while conversion job with this job ID is running.

This workflow action is used to wait while specified conversion job is running. Note that conversion job may take from few minutes to several hours depends on different factors.

See Also: "Convert Document" and "Get Conversion Job Status" actions.

Parameters:

with this job ID - ID of conversion job returned by "Convert Document", "Convert Folder at Url" or "Convert Library" actions.

Examples:

Convert document Current Item to PDF format and store job ID in Variable: ConversionJobId. Overwrite existing document.

then Wait while conversion job Variable: ConversionJobId is running

then Get conversion job Variable: ConversionJobId status and store in Variable: JobStatus; store links to converted documents in Variable: DocumentsArray

then Get item at 0 index from Variable: DocumentsArray and store in Variable: FileName

then Log Status: [%Variable: JobStatus%]. File name: [%Variable: FileName%] to the workflow history list

Result in workflow history list:

Status: Completed. File name: http://sp14.vlab/Shared Documents/Invoice 1730A.pdf

7. Document Library activities

7.1 Add Watermark To PDF Document

Insert a watermark (text or image url) in a PDF file at this url. Save result to the folder at specified url and with the same name. Overwrite existing document.

This workflow action is used to add a text watermark or image watermarks to a PDF file.

Seel also: Use the "Set PDF Watermark Options" action prior this action to set the advanced options for the watermark.

Parameters:

text - watermark text.
image url - watermark picture url.
Either of these parameters should be specified. If both parameters specified, **image url** parameter will be used.

PDF file at this url - specify the URL of the PDF document.

specified url - specify the URL of the folder to save the output PDF (with watermark added)

the same name - specify the name for the output PDF, it could be the same name, or any other custom name.

Overwrite - choose to Overwrite/Do not overwrite if the document with this name already exists in the target folder.

7.2 Check In Item at Url

Do minor check in of list item at specified url with specified comment

This workflow action is used to check in specified item with a comment.

Parameters:

minor check in - Check in type, one from following options: *minor check in, major check in, overwrite check in*.

specified url - Full list item URL including site name.

specified comment - Comment text.

7.3 Check Out Item at Url

Check out item at specified url

This workflow action is used to check out list item specified by URL.

Parameters:

specified url - Full URL of list item including site name.

7.4 Copy Document to List Item Attachment

Copy document at this url to list item at this url. Overwrite existing items

This workflow action copies the specified document to a list item as an attachment.

Parameters:

Copy document at **this url** - the URL of the document to be copied
to list item at **this url** - the URL of the target list item
Overwrite - select to overwrite or not the existing items

7.5 Copy or Move Document

Copy document at this url to the folder at specified url and with the same name. Overwrite existing document. Trigger events.

This workflow action is used to copy, move or rename documents in SharePoint document libraries.

It can also **copy** a document to **multiple folders**, if they are specified in an array.

Parameters:

Copy - Copy or move document.
this url - URL of the document at SharePoint Server.
specified url - URL of the target folder, may be the same as source folder.
Arrays of URLs are also supported for **copying** a document to all folders mentioned in an array.
the same - Optional parameter, new file name and extension for the document.
Overwrite - If the folder already contains file with the same name, then you can apply one from two options. *Overwrite* - replace file by new one. *Do not overwrite* - skip this file.
Note: This action adds new file version when overwriting
Trigger events - select *Trigger* or *Do not trigger* events for this action.

Examples:

Move document at http://gorlach7/Shared%20Documents/1.png to the folder at http://gorlach7/Shared%20Documents/ and with 2.png name. Overwrite existing document.

This sample workflow renames 1.PNG file to 2.PNG.

7.6 Copy or Move Document to Different Farm

Copy document at this url to the different farm's folder at specified url and with the same name. Overwrite existing document.

This workflow action is used to copy or move specified document to the document library located on a different SharePoint farm.

Important Note: the action *'Set Target Site Options'* always should be used before this action!

Additionally you can use '*Set Encrypted Passwords*' action and *Encrypt Credentials* tool to specify the credentials in secure way.

See Also: 'Set Encrypted Passwords' and 'Set Target Site Options' actions.

Parameters:

Copy - Specifies whether to copy or move document.

this url - Full URL of the source document.

specified url - Full URL of the target document library on a different SharePoint farm.

the same name - Optional parameter, new file name and extension for the document.

Overwrite - If the document library already contains document with the same name, then you can apply one from two options. *Overwrite* - replace document with the new one. *Do not overwrite* - skip this document.

Note: This action adds new file version when overwriting

7.7 Copy or Move Folder

Copy all documents of this content type from a folder at this url to the folder at this url. Overwrite existing documents. Include subfolders.

This workflow action copies or moves the files from the specified folder to another folder. Can optionally include files in subfolders.

Parameters:

Copy/Move - select to copy or move files.

this content type - (Optional) Use this optional parameter to copy/move documents only with the indicated content type.

By default, all documents of all content types are copied/moved.

folder at this url - specify the URL of the source folder

to the folder at this url - specify the URL of the target folder

Overwrite/Do not overwrite - specify if the existing files with the same names in the target folder should be overwritten, or the workflow should stop with an error.

Include/Do not include subfolders - specify if the subfolders in the source folder should also be copied/moved.

7.8 Create Document from File

Create document from file path in parent folder url. Overwrite existing document. Trigger events.

This workflow action is used to add new document or replace existing document by file from disk or network.

Note: In order to specify file access credentials, add the "Set Network File Access" activity to the same workflow step first.

Parameters:

file path - Disk file path or network file path.

parent folder url - Full URL including site name of document library of the folder in document library.

Overwrite - If the folder already contains file with the same name, then you can apply one from three options. *Overwrite* - replace file in library by specified file. *Do not overwrite* - generate an error. *Append unique identifier* - add string like _20e0ad06-4261-4349-8a37-894fa7574f42 to the end of file name.

Trigger - Specifies whether to trigger an item create event.

7.9 Create Document from Text

Create document from this text in file url. Overwrite existing document. Trigger events.

This workflow action is used to add new text document or replace existing document by new file created from string variable.

Parameters:

this text - Document body.
file url - Full URL for the new document, including site name, file name and extension.
Overwrite - If the folder already contains file with the same name, then you can apply one from three options. *Overwrite* - replace file in library by specified file. *Do not overwrite* - generate an error. *Append unique identifier*- add string like _20e0ad06-4261-4349-8a37-894fa7574f42 to the end of file name.
Trigger - Specifies whether to trigger an item create event.

7.10 Create Document Set in the Document Library

Create this title document set with this description based on this content type in the folder at this url

This workflow action creates a document set in a specified document library.

Note: Since the document set from technical point of view behaves the same way as a document library, you can use the actions fordocument libraries to manage the document set.

Parameters:

this title - the document set name
this description - the text description of the document set
this content type - the name of the document set content type that new document set should be based on
folder at this url - the URL of the folder in a document library where the document set will be created

7.11 Create Folder in the Document Library

Create specified folder in the current folder.

This workflow action is used to create folder in document library.

Parameters:

specified - Name of the folder to create.
current/root - Specifies where to create a new folder: in the current folder of document library or in the root folder.

7.12 Create Folder in the Document Library at Url

Create specified folder at folder url.

This workflow action is used to create subfolder in document library or in folder of document library.

Parameters:

specified - Folder name.

url - Full URL with site name of document library or folder in document library.

7.13 Create Link to a Document

Create link to document with this url at this library url

This workflow action creates a link in specified library to a document from another library

Parameters:

this url - the url of the document for which the link will be created

this library url - URL of the library where the link will be created

7.14 Delete Folder in the Document Library

Delete specified folder in the current folder.

This workflow action is used to delete specified folder with all its content in document library. The error happens if the folder is not found.

Parameters:

specified - Name of the folder to delete.

current/root - Specifies where to delete the folder: in the current folder of document library or in the root folder.

7.15 Delete Folder in the Document Library at Url

Delete specified folder at folder url

This workflow action is used to delete subfolder in document library or in folder of document library.

Parameters:

specified - Subfolder name to delete.

url - Full URL with site name of document library or name of the folder in the document library.

7.16 Discard Check Out Item at Url

Discard Check Out item at specified url

This workflow action is used to discard check out of specified list item.

Parameters:

specified url - Full list item URL including site name.

7.17 Discard Publish Item

Discard publish list item at this url with specified comment

This workflow action is used to discard list item publishing.

Parameters:

this url - Full URL of target list item.

specified comment - Comment text.

7.18 Download Documents to Folder

Recursively download all documents from this folder url to specified path. Overwrite existing files.

This workflow action is used to save all documents from document library or folder of document library to local disk or to network share.

Note: In order to specify file access credentials, add the "Set Network File Access" activity to the same workflow step first.

Parameters:

Recursively - One from two options. *Recursively*: save the document library or folder content including all subfolders, subfolders tree will be created in *specified path*. *Non-recursively*: save files only from specified document library of folder.
all - File mask, like *.jpg. You can specify only one file mask.
this folder url - Document library or folder of document library.
specified path - Local disk or network path.
Overwrite - If the folder already contains file with the same name, then you can apply one from four options. *Overwrite* - replace file in library by new file. *Do not overwrite* - generate an error. *Skip* - do not overwrite file in library without error. *Append unique identifier for* - add string like _20e0ad06-4261-4349-8a37-894fa7574f42 to the end of file name.

7.19 Get Document Content

Get content of the document at this url and store in Variable: variable.

This workflow action is used to retrieve contents of specfied document in to output variable.

Parameters:

this url - URL of the document at SharePoint Server.
Variable: variable - Output variable.

7.20 Get Entry From Folder Property Bag

Get specified entry value from list item at this url parent folder property bag and store in Variable: variable

This workflow action is used to get specified entry value from property bag of parent folder of specified list item.

For the list of property bag entries see: http://msdn.microsoft.com/en-us/library/microsoft.sharepoint.spfolder.properties(v=office.12).aspx

Error happens if property is not found.

Parameters:

specified - Name of the entry in property bag, e.g. vti_timelastmodified.
this url - Full URL of list item.
Variable: variable - Output variable.

7.21 Publish Item

Publish list item at this url with specified comment

This workflow action is used to submit the list item for content approval with the specified comment.

Parameters:

this url - Full URL of target list item.
specified - Comment text.

7.22 Rename Document at Url

Rename document at specified url to this file name

This workflow action is used to rename list item specified by URL.

Parameters:

specified url - Full URL of list item including site name.
this file name - New file name without extension.

7.23 Save Document to File

Save list item at this url to output folder. Overwrite existing file.

This workflow action is used to save document from documents library to local disk or network share.

Note: In order to specify file access credentials, add the "Set Network File Access" activity to the same workflow step first.

Parameters:

this url - Full URL of list item.
output folder - Local disk folder or network share.
Overwrite - If the folder already contains file with the same name, then you can apply one from four options. *Overwrite* - replace file in library by new file. *Do not overwrite* - generate an error. *Skip* - do not overwrite file in library without error. *Append unique identifier for* - add string like _20e0ad06-4261-4349-8a37-894fa7574f42 to the end of file name.

Examples:

Save [%Current Item:Encoded Absolute URL%] to C:\temp\. Overwrite existing file.

7.24 Set Network File Access

Set network file access using this user and this password.

This workflow action is used to specify credentials for a group of Document Library activities that works with files:

- Save Document to File
- Create Document From File
- Upload Documents From Folder
- Download Documents To Folder
- Upload Files To List Item Attachments

Note: The password will be stored as a plain text in the workflow XOML file and may be accessed by the URL like *http://sharepoint/Workflows/workflowname/workflowname.xoml* or by SharePoint Designer.

Parameters:

this user - User name that will be used for file access (like DOMAIN\Account).
this password - Password that will be used for file access (see Note).

Examples:

Set network file access using HAREPOINT\testuser and pwd123.

7.25 Set PDF Watermark Options

Set watermark Rotation: None, Opacity: 50, Page range: all. Font name: Courier, size: 53, color: 00000000, style: Regular.

This workflow action sets the options for the watermark that will be added to a PDF document by an "Add Watermark To PDF Document" action. These actions need to be used in combination.

See also: "Add Watermark To PDF Document" action.

Parameters:

Rotation: None - available values:
Diagonal - rotated 45 degrees clockwise **ReverseDiagonal** - rotated 45 degrees counterclockwise **none** - no rotation
Opacity: 50 - opacity in percents from 0% to 100%.
Page range: all - page range where watermarks should be added. Possible values: **all** - for all pages **1-5, 8, 11-13** - to print pages 1,2,3,4,5,8,11,12,13
Font name: Courier - font family name
Note: this parameter takes effect only for **text** watermarks
size: 53 - font size in points
Note: this parameter takes effect only for **text** watermarks
color: FF000000 - black color in 16-bit ARGB color space
Note: this parameter takes effect only for **text** watermarks

Examples:

style: Regular - font style.

Available values: **Regular, Bold, Italic, Underline, Strikeout**.

Note: this parameter takes effect only for **text** watermarks

7.26 Set Target Site Options

Use site at this url url as target site. Logon using this user user name and this password password.

This workflow action is only used to set target SharePoint site URL, login and password for actions that work with remote SharePoint farm like "*Copy or Move Document to Different Farm*" or *"Copy List Item to Different Farm*".

This action must be placed above all other activities that can use stored options and credentials. You can use this action several times in one workflow step to use different target SharePoint farms.

Note: This workflow action **will not** take effect on other actions, except actions that work with remote SharePoint farm like "*Copy or Move Document to Different Farm*" or *"Copy List Item to Different Farm*".

Note: The password will be stored as a plain text in the workflow XOML file and may be accessed by the URL like*http://sharepoint/Workflows/workflowname/workflowname.xoml* or by SharePoint Designer.

Parameters:

this url - Full URL of the target site on the different SharePoint farm.
this user - User name that will be used to access remote SharePoint farm.
this password - Password that will be used to access remote SharePoint farm.

7.27 Upload Documents from Folder

Recursively upload all files from specified path to this folder url. Overwrite existing documents. Trigger events.

This workflow action is used to upload files from local disk or network share to document library.

Note: In order to specify file access credentials, add the "Set Network File Access" activity to the same workflow step first.

Parameters:

Recursively - One from two options. *Recursively*: upload files from specified path including all subfolders, subfolders tree will be created in specified document library. *Non-recursively*: upload files only from specified path.

all - File mask, like *.jpg. You can specify only one file mask.

specified path - Disk or network path.

this folder url - Document library or folder of document library.

Overwrite - If the folder already contains file with the same name, then you can apply one from four options. *Overwrite* - replace file in library by new file. *Do not overwrite* - generate an error. *Skip* - do not overwrite file in library without error. *Append unique identifier for* - add string like _20e0ad06-4261-4349-8a37-894fa7574f42 to the end of file name.

Trigger - Specifies whether to trigger an item create/update event.

Examples:

Recursively upload all files from C:\ to http://mysite/Shared%20Documents/. Overwrite existing documents

Never try this example.

8. E-Mail activities

8.1 Add Custom Headers To Email

Add these custom headers to outgoing e-mail messages

This workflow action adds the specified e-mail headers to the e-mails sent by the HarePoint *Send E-Mail* actions.

Parameters:

these custom headers - specify the headers to be added. Multple headers can be specified, delimited by a new line.

Header examples:

- *Return-Receipt-To: sender@example.com* - request a receipt
- *Importance: high* - set importance level as High
- *Sensitivity: company-confidential* - set the e-mail sensitivity as confidential

For more details, please refer to https://tools.ietf.org/html/rfc4021

8.2 Get E-mail Property

Get HTML body from this mime body and store in Variable: variable.

This workflow action is used to retrieve message properties from the mime body. The list of available properties: HTML body, TEXT body, sender name, sender e-mail, CC recipients, reply to, date sent, importance, mail headers.

See Also: "Receive E-Mail from POP3 Server" and "Receive E-Mail from IMAP Server" actions.

Parameters:

HTML body - One of the available message properties.

this mime - Mime body of the received e-mail message .

Variable: variable - String value of the specified property .

Examples:

Get HTML body from Variable: mimeBody body and store in Variable: variable.

8.3 Receive All E-Mails to List from IMAP Server

Download all e-mails from this server, secure connection type None, login this login and password this password. Store them with attachments into the list at this url. Store inline images in folder at this url.

This workflow action is used to download all messages from the specified IMAP mailbox from the default folder (usually INBOX) to the prepared SharePoint list. The messages are deleted from the mailbox after adding into SharePoint list.

The target SharePoint list should contain the following fields, that will be filled by the values from the downloaded messages:

Column	Data Type	E-mail header
Title	Single line of text	Subject
Description	Multiple lines of text / enhanced rich text	Html Body
FromName	Single line of text	Sender display name
FromAddress	Single line of text	Sender e-mail address
FromUser	Person or Group	The program tries to find SharePoint user with the message sender e-mail address.
Date	Date and Time	Date Sent
Importance	Single line of text	Message importance: Low, Normal, High
TextBody	Multiple lines of text	Text body of the message
ToString	Single line of text	"To" field with names and e-mail addresses
CCString	Single line of text	"CC" field with names and e-mail addresses
ReplyToString	Single line of text	"Reply-To" header
MailHeaders	Multiple lines of text	All message headers

Parameters:

this server - IMAP server name or IP-address. By default port 143 is used, you can specify port with colon, like mail.company.com:1143.
type **None** - Secure connection type. Available values: *None, SSL*. Note, that IMAP server with SSL usually listen for incoming connections at port 993.
this login - User name for the IMAP server.
this password - Password for the IMAP server.
with - Specify whether to attach message attachments to list item or not.
this url - Full URL of the list.
this url - (Optional) Full URL of the document library where inline attachments will be stored.

Examples:

Download all e-mails from imap.mailserver.com:143, secure connection type None, login harepoint@gmail.com and password pwd123. Store them with attachments into the list at Variable: listUrl. Store inline images in folder at Variable: folderUrl.

8.4 Receive All E-Mails to List from POP3 Server

Download all e-mails from this server, secure connection type None, login this login and password this password. Store them with attachments into the list at this url. Store inline images in folder at this url.

This workflow action is used to download all messages from the specified POP3 mailbox to the prepared SharePoint list. The messages are deleted from the mailbox after adding into SharePoint list.

The target SharePoint list should contain the following fields, that will be filled by the values from the downloaded messages:

Column	Data Type	E-mail header
Title	Single line of text	Subject
Description	Multiple lines of text / enhanced rich text	Html Body
FromName	Single line of text	Sender display name
FromAddress	Single line of text	Sender e-mail address
FromUser	Person or Group	The program tries to find SharePoint user with the message sender e-mail address.
Date	Date and Time	Date Sent
Importance	Single line of text	Message importance: Low, Normal, High
TextBody	Multiple lines of text	Text body of the message
ToString	Single line of text	"To" field with names and e-mail addresses
CCString	Single line of text	"CC" field with names and e-mail addresses
ReplyToString	Single line of text	"Reply-To" header
MailHeaders	Multiple lines of text	All message headers

.

Parameters:

this server - POP3 server name or IP-address. By default port 110 is used, you can specify port with colon, like mail.company.com:1110.
type **None** - Secure connection type. Available values: None, SSL. Note, that POP3 server with SSL usually listen for incoming connections at port 995.
this login - User name for the POP3 server.
this password - Password for the POP3 server.
with - Specifies whether to attach message attachments to list item or not.
this url - Full URL of the list.
this url - (Optional) Full URL of the document library where inline attachments will be stored.

Examples:

Download all e-mails from pop.gmail.com:995, secure connection type SSL, login harepoint@gmail.com and password pwd123. Store them with attachments into the list at Variable: listUrl. Store inline images in folder at Variable: folderUrl.

8.5 Receive E-Mail from IMAP Server

Receive email from this server, secure connection type None, login this login and password this password. Store e-mail sender in Variable:variable1, recipient in Variable:variable2, subject in Variable:variable3 and HTML body in Variable:variable4.

This workflow action is used to receive e-mail message from IMAP mail server from a default folder (usually INBOX). Action gets the list of messages in mailbox, receives the first one and deletes the received message from mailbox.

Note: The password will be stored as a plain text in the workflow XOML file and may be accessed by the URL like *http://sharepoint/Workflows/workflowname/workflowname.xoml* or by SharePoint Designer.

See Also: "Save E-Mail Attachments to List Item" and "Get E-Mail Property" actions.

Parameters:

this server - IMAP server name or IP-address. By default port 143 is used, you can specify port with colon, like mail.company.com:1430.
type **None** - Secure connection type. Available values: *None, SSL*. Note, that IMAP server with SSL usually listen for incoming connections at port 993.
this login - User name for the IMAP server.
this password - Password for the IMAP server.
Variable: variable1 - Sender address of received e-mail message.
Variable: variable2 - Recipients addresses of received e-mail message.
Variable: variable3 - Subject of received e-mail message.
Variable: variable4 - Body of received e-mail message. The type of body is specified by the previous parameter. It can be one of the following: HTML, TEXT or MIME.

Examples:

Receive email from imap.mailserver.com, secure connection type None, login harepoint@mailserver.com and password pwd123. Store e-mail sender in Variable: sender, recipient in Variable: recipient, subject in Variable: subject and HTML body in Variable: body.

8.6 Receive E-Mail from POP3 Server

Receive email from this server, secure connection type None, login this login and password this password. Store e-mail sender in Variable:variable1, recipient in Variable:variable2, subject in Variable:variable3 and HTML body in Variable:variable4.

This workflow action is used to receive e-mail message from POP3 mail server. Action gets the list of messages in mailbox, receives the first one and deletes the received message from mailbox.

Note: The password will be stored as a plain text in the workflow XOML file and may be accessed by the URL like *http://sharepoint/Workflows/workflowname/workflowname.xoml* or by SharePoint Designer.

See Also: "Save E-Mail Attachments to List Item" and "Get E-Mail Property" actions.

Parameters:

this server - POP3 server name or IP-address. By default port 110 is used, you can specify port with colon, like mail.company.com:1110.
type **None** - Secure connection type. Available values: *None, SSL*. Note, that POP3 server with SSL usually listen for incoming connections at port 995.
this login - User name for the POP3 server.
this password - Password for the POP3 server.
Variable: variable1 - Sender address of received e-mail message.
Variable: variable2 - Recipients addresses of received e-mail message.
Variable: variable3 - Subject of received e-mail message.
Variable: variable4 - Body of received e-mail message. The type of body is specified by the previous parameter. It can be one of the following: HTML, TEXT or MIME.

Examples:

Receive email from pop.gmail.com:995, secure connection type SSL, login harepoint@gmail.com and password pwd123. Store e-mail sender in Variable: sender, recipient in Variable: recipient, subject in Variable: subject and HTML body in Variable: body.

8.7 Save E-Mail Attachments to List Item

Save all attachments from this mime body to list item at this url. Store names of saved attachments to Variable: array.

This workflow action is used to retrieve attachments from the e-mail message and attach them to the specified list item.

See Also: "Receive E-Mail from POP3 Server" and "Receive E-Mail from IMAP Server" actions.

Parameters:

all - File masks, separated by semicolon. Leave this field empty to save all attachments.
this mime - String with the message mime body.
this url - Full URL of list item.
Variable: array - Array of the saved attachment names.

Examples:

Save *.jpg;*.png attachments from Variable: mimeBody body to list item at Variable: itemUrl. Store names of saved attachments to Variable: array.

8.8 Send E-Mail with Document as Attachment

Send e-mail to this address with Normal importance, blind carbon copy to this address, display this address as sender and include document this document as attachment in the e-mail.

This workflow action is used to send e-mail with the document from document library as attachment.

By default, farm settings for outgoing emails (Central Administration \ Outgoing E-Mail Settings) will be used.

See Also: "Set SMTP Server Options" action.

Parameters:

e-mail to **this address** - Click here to specify message recipients, carbon copy recipients, message subject and body.
Normal - E-mail importance. Available options: *Normal, Low, High*.
blind carbon copy to **this address** - Blind carbon copy recipients, maybe blank.
this address as sender - Sender address. If this value is not specified then value from farm settings for outgoing emails will be used.
this document - Item from document library. For example, *Current Item* for the document library workflow.

8.9 Send E-Mail with File Attachments

Send e-mail to this address with Normal importance, blind carbon copy to this address, use this address as sender and include a file this file as attachment.

This workflow action is used to send e-mail with one or more file attachments from local disk or network share.

By default, farm settings for outgoing emails (Central Administration \ Outgoing E-Mail Settings) will be used.

See Also: "Set SMTP Server Options" action.

Parameters:

e-mail to **this address** - Click here to specify message recipients, carbon copy recipients, message subject and body.
Normal - E-mail importance. Available options: *Normal, Low, High*.
blind carbon copy to **this address** - Blind carbon copy recipients, maybe blank.
this address as sender - Sender address. If this value is not specified then value from farm settings for outgoing emails will be used.
this file - Multi-line string with one or more file names from local disk or network share to attach. Each file name must be on a separate line.

Examples:

Use SMTP server exchange.local:125, secure connection type None, login this login and password this password.

then Send e-mail to gorlach@exchange.local with Normal importance, blind carbon copy to this address, use noreply@exchange.local as sender and include a file C:\temp\report.pdf as attachment.

8.10 Send E-Mail with Http File Attachment

Send e-mail to this address with Normal importance, blind carbon copy to this address, use this address as sender and include a file from this url as attachment named file name.

This workflow action is used to send e-mail with file attachment specified by URL.

By default, farm settings for outgoing emails (Central Administration \ Outgoing E-Mail Settings) will be used.

See Also: "Set SMTP Server Options" action.

Parameters:

e-mail to **this address** - Click here to specify message recipients, carbon copy recipients, message subject and body.
Normal - E-mail importance. Available options: *Normal, Low, High*.
blind carbon copy to **this address** - Blind carbon copy recipients, maybe blank.
this address as sender - Sender address. If this value is not specified then value from farm settings for outgoing emails will be used.
this URL - Full URL to file.
file name - Optional new name and extension for attached file, leave blank to use original name.

Examples:

Send e-mail to harepoint@gmail.com with Normal importance, blind carbon copy to this address, use noreply@harepoint.com as sender and include a file from Workflow Context:Current Item URL as attachment named file name.

8.11 Send E-Mail with InfoPath Attachments

Send e-mail to this address with Normal importance, blind carbon copy to this address, use this address as sender and add InfoPath attachment specified by XPATH of list item at this url to attachments of e-mail.

This workflow action is used to send e-mail with InfoPath attachments of list item.

By default, farm settings for outgoing emails (Central Administration \ Outgoing E-Mail Settings) will be used.

See Also: "Set SMTP Server Options" action.

Parameters:

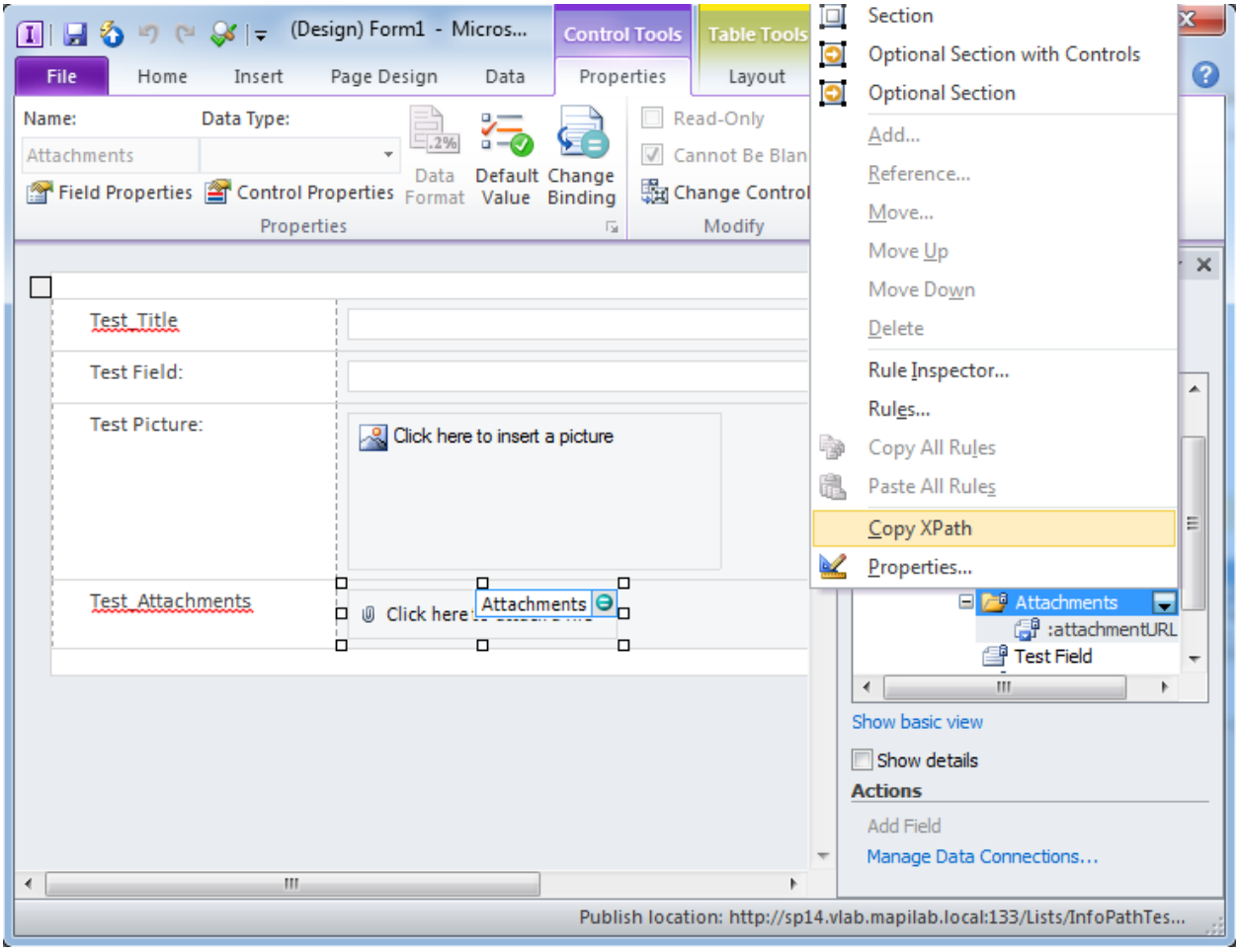
e-mail to **this address** - Click here to specify message recipients, carbon copy recipients, message subject and body.
Normal - E-mail importance. Available options: Normal, Low, High.
blind carbon copy to **this address** - Blind carbon copy recipients, maybe blank.
this address as sender - Sender address. If this value is not specified then value from farm settings for outgoing emails will be used.
specified by XPATH - XPath query string. See examples section for details.
this item - InfoPath attachments of this list item will be attached to e-mail.

Examples:

Send e-mail to harepoint@gmail.com with Normal importance, blind carbon copy to this address, use noreply@harepoint.com as sender and add InfoPath attachment /dfs:myFields/dfs:dataFields/my:SharePointListItem_RW/my:Attachments of list item at Variable: thisUrl to attachments of e-mail.

You can combine multiple XPath queries using vertical bar ('|') sign, e.g. /my:myFields/my:MoreAttachments | /my:myFields/my:AttachFile.

You can retrieve XPath string using Microsoft InfoPath Designer:



8.12 Send E-Mail with InfoPath Form

Send e-mail to this address with Normal importance, blind carbon copy to this address, use this address as sender and attach InfoPath form at this url

This workflow activity sends an e-mail containing InfoPath form that will be displayed in Outlook, and also upon double-click it will be opened in InfoPath Form Filler.

Note: this action works for form libraries only.

By default, farm settings for outgoing emails (Central Administration \ Outgoing E-Mail Settings) will be used.

See Also: "Set SMTP Server Options" action.

Parameters:

this address - specify recipient's e-mail address

Normal importance - specify importance level. Available levels: Normal, High, Low
blind carbon copy to **this address** - (optional) specify blind carbon copy recipient's address
this address as sender - specify e-mail address that will appear as e-mail sender address
InfoPath form at **this url** - specify the URL of the InfoPath form to be sent

8.13 Send E-Mail with List Item Attachments

Send e-mail to this address with Normal importance, blind carbon copy to this address, use this address as sender and include all files attached to this item to attachments of e-mail.

This workflow action is used to send e-mail with attachments of list item. For example, you can send attachments of Task or Calendar item with this action.

By default, farm settings for outgoing emails (Central Administration \ Outgoing E-Mail Settings) will be used.

See Also: "Set SMTP Server Options" action.

Parameters:

e-mail to **this address** - Click here to specify message recipients, carbon copy recipients, message subject and body.
Normal - E-mail importance. Available options: *Normal, Low, High*.
blind carbon copy to **this address** - Blind carbon copy recipients, maybe blank.
this address as sender - Sender address. If this value is not specified then value from farm settings for outgoing emails will be used.
this item - Attachments of this list item (like Calendar or Task) will be attached to e-mail.

8.14 Send E-Mail with Multiple Attachments

Send e-mail to this address with Normal importance, blind carbon copy to this address, use this address as sender and add all files at this urls to attachments of e-mail.

This workflow action is used to send e-mail with multiple file attachment specified by array of URLs.

By default, farm settings for outgoing emails (Central Administration \ Outgoing E-Mail Settings) will be used.

See Also: "Set SMTP Server Options" action.

Parameters:

e-mail to **this address** - Click here to specify message recipients, carbon copy recipients, message subject and body.
Normal - E-mail importance. Available options: *Normal, Low, High*.
blind carbon copy to **this address** - Blind carbon copy recipients, maybe blank.
this address as sender - Sender address. If this value is not specified then value from farm settings for outgoing emails will be used.
this urls - Array of full URLs to files to be attached.

Examples:

Find item in the list at http://spsite/Shared%20Documents where item's Name is not empty value . Store result urls in Variable: array .

Use SMTP server vpcberlin, secure connection type None , login test and password test . Do not convert message body to plain text.

then Send e-mail to Sample User with Normal importance, blind carbon copy to this address , use test@mapilab.local as sender and add all files at Variable: array to attachments of e-mail.

8.15 Send E-Mail without Attachments

Send e-mail to this address with Normal importance, blind carbon copy to this address, display this address as sender.

This workflow action is used to send e-mail without any attachments.

By default, farm settings for outgoing emails (Central Administration \ Outgoing E-Mail Settings) will be used.

See Also: "Set SMTP Server Options" action.

Parameters:

e-mail to **this address** - Click here to specify message recipients, carbon copy recipients, message subject and body.
Normal - E-mail importance. Available options: *Normal, Low, High*.
blind carbon copy to **this address** - Blind carbon copy recipients, maybe blank.
this address as sender - Sender address. If this value is not specified then value from farm settings for outgoing emails will be used.

8.16 Send Meeting Request

Send meeting request to this address, meeting starts at this time, ends at this time, location at this location, request responses: yes

This workflow action is sending meeting request

Note: Prior to this action, 'Set Exchange Web Service Options' action should be used.

Parameters:

this address - recipient's e-mail address
starts at **this time** - the meeting starting time
ends at **this time** - the meeting ending time, leave blank for all day long event
this location - location where the meeting will be held
request responses: **yes** - a value indicating whether responses are requested when invitations are sent for this meeting

8.17 Send SMS Message

Send SMS message to this phone numbes with this text

This workflow action sends SMS messages to specified phone numbers with a specified text.

SMS settings should be set up at Central Administration -> System Settings -> Mobile Account

An example for RedOxygen provider:
http://www.redoxygen.com/support/wiki/doku.php id=sharepoint:integration

Parameters:

this phone numbers - text string with the phone numbers new line delimited, or the array. Should contain digits only, other symbols will be removed automatically.

this text - the text of the SMS message. The length is not limited, it will be sent as several SMS messages automatically if necessary.

Examples:

Send SMS Message to 446582513 with Hello! text

8.18 Set SMTP Server Options

Use SMTP server this server, secure connection type None, login this user and password this password. Do not convert message body to plain text.

This workflow action is used to set SMTP server settings for HarePoint SMTP-related activities like "Send E-Mail with List Item Attachments". This activity must be placed above all other SMTP-related activities in the current workflow step.

Otherwise, farm settings for outgoing emails (Central Administration \ Outgoing E-Mail Settings) will be used for SMTP-related HarePoint activities.

Note: The password will be stored as a plain text in the workflow XOML file and may be accessed by the URL like*http://sharepoint/Workflows/workflowname/workflowname.xoml* or by SharePoint Designer.

Parameters:

this server - SMTP server name or IP-address. By default port 25 is used, you can specify port with colon, like mail.company.com:125.

type **None** - Secure connection type. Available values: *None, SSL*. Note, that SMTP server with SSL usually listen for incoming connections at port 465.

this user - Login for SMTP server, may be blank.

this password - Password for SMTP server, may be blank.

Do not convert - Specifies whether to convert message body to plain text or not.

Examples:

Use SMTP server exchange.local:125, secure connection type None, login this login and password this password. Do not convert message body to plain text.

then Send e-mail to gorlach@exchange.local with Normal importance, blind carbon copy to this address, use noreply@exchange.local as sender and include a file C:\temp\report.pdf as attachment.

9. Images activities

9.1 Convert Image

Convert image at specified url to JPG format and store URL to converted image in Variable: ConvertedImageUrl. Overwrite existing image.

This workflow action converts image to one of supported output formats: Bitmap Picture (bmp), GIF, JPG, PNG, TGA, TIFF.

Output file will have the same name and the extension according your selection in the second parameter, and will be placed in the same document library or folder where the source file is stored.

Note: See 'Get Image Size' activity overview for the list of supported file formats.

Parameters:

specified url - Full URL of the picture in Document Library.
JPG - One of supported output formats: Bitmap Picture (bmp), GIF, JPG, PNG, TGA, TIFF.
Variable: ConvertedImageUrl - Output string variable with the URL of a new picture.
Overwrite - Choose overwrite the file or do not, if the document library already contains a file with the same name as output file.

Examples:

If image at Current Item:Encoded Absoulte URL is supported

If image at Current Item:Encoded Absoulte URL uses compression

Log Conversion is not required to the workflow history list

else

Convert image at Current Item:Encoded Absoulte URL to JPG format and store URL of converted image in Variable: ConvertedImageUrl. Overwrite existing image.

then Delete item at Current Item:Encoded Absoulte URL list item url.

This workflow automatically converts uncompressed images to JPG format.

9.2 Convert Images in Folder

Convert images in folder at this url to JPG format and store converted images in specified url. Overwrite existing image. Do not delete source images

This workflow action is used to convert all images of supported formats in specified Document Library or folder of Document Library. Source and target folders or Document Libraries may be the same.

Pay attention that source images which already have a type specified by second parameter will not be copied to the target folder. Use built-in "Copy List Item" activity if required.

Note: See 'Get Image Size' activity overview for the list of supported file formats.

Parameters:

this url - Full URL of Document Library or folder of Document Library with source files.
JPG - One of supported output formats: Bitmap Picture (bmp), GIF, JPG, PNG, TGA, TIFF.
specified url - Full URL of target Document Library or folder of Document Library to store converted images. May be the same as source URL.
Overwrite - Overwrite or do not overwrite pictures in target Document Library.
Do not delete - Delete or do not delete source images.

9.3 Convert Images in Library

Convert images in this library to JPG format and store converted images in specified library. Overwrite existing image. Do not delete source images

This workflow action is used to convert all images of supported formats in specified Document Library. Source and target Document Libraries may be the same.

Pay attention that source images which already have the type specified by the second parameter will not be copied to target Document Library. Use built-in "Copy List Item" activity if required.

Note: See 'Get Image Size' activity overview for the list of supported file formats.

Parameters:

this library - Source Document Library.

JPG - One from supported output formats: Bitmap Picture (bmp), GIF, JPG, PNG, TGA, TIFF.

specified library - Target Document Library. Note that source and target Document Libraries may be the same.

Overwrite - Overwrite or do not overwrite pictures in target Document Library.

Do not delete - Delete or do not delete source images.

9.4 Flip Image

Flip image at specified url horizontally.

This workflow action is used to flip image in one from supported formats horizontally or vertically.

Note: See 'Get Image Size' activity overview for the list of supported file formats.

Parameters:

specified url - URL of the image in Document Library.

horizontally - How to flip the image: horizontally or vertically.

9.5 Get EXIF Property from Image

Get EXIF property Title from image at this url and store in Variable: variable, raw value in Variable: variable1.

This workflow action is used to get EXIF property from an image in one from supported formats. With this action you can get the following properties:

- Title
- Camera Manufacturer
- Camera Model
- Exposure Time
- F number

- Date and Time
- Focal Length
- Orientation
- Exposure program
- ISO speed rating
- Flash
- Exposure mode
- Artist
- White balance
- Scene capture type
- Exposure bias
- Aperture
- Maximum lens aperture
- North or South Latitude
- East or West Longitude
- Scene type
- Software

See also: Exchangeable image file format at Wikipedia
(http://en.wikipedia.org/wiki/Exchangeable_image_file_format).

Note: See 'Get Image Size' activity overview for the list of supported file formats.

Parameters:

Title - Name of the EXIF property (see list above).
this url - URL of the picture in Document Library.
Variable: variable - Output string variable. Empty string if the property is not found.

Variable: variable - Output string variable with the value in the form as it is stored in the image.. Empty string if the property is not found.

9.6 Get Image Size

Get size of the image in specified url and store width in Variable: ImageWidth and height in Variable: ImageHeight

This workflow action is used to get image size in pixels. In case of an unknown format of the file, this action will generate an error.

File formats supported by the Product image processing engine:

- BMP files [reading, writing]
- DDS files [reading]
- EXR files [reading, writing]
- Raw Fax G3 files [reading]
- GIF files [reading, writing]
- HDR files [reading, writing]
- ICO files [reading, writing]
- IFF files [reading]
- JNG files [reading]
- JPEG/JIF files [reading, writing]
- JPEG-2000 File Format [reading, writing]
- JPEG-2000 codestream [reading, writing]
- KOALA files [reading]
- Kodak PhotoCD files [reading]
- MNG files [reading]
- PCX files [reading]
- PBM/PGM/PPM files [reading, writing]

- PFM files [reading, writing]
- PNG files [reading, writing]
- Macintosh PICT files [reading]
- Photoshop PSD files [reading]
- RAW camera files [reading] (see note below)
- Sun RAS files [reading]
- SGI files [reading]
- TARGA files [reading, writing]
- TIFF files [reading, writing]
- WBMP files [reading, writing]
- XBM files [reading]
- XPM files [reading, writing]

Note: Product can load RAW files from digital photo cameras (CRW/CR2, NEF, RAF, DNG, MOS, KDC, DCR, etc), virtually all RAW formats are supported).

Parameters:

specified url - Full URL of the picture in Document Library.
Variable: ImageWidth - Output variable for image width in pixels.
Variable: ImageHeight - Output variable for image height in pixels.

9.7 Get XMP Property

Get specified XMP metadata property from document at this url and store result in Variable: variable1

This workflow action gets the specified XMP property from the specified document and saves it to a variable.

Parameters:

specified XMP metadata - select the required XMP metadata from the drop-down list
document at this url - specify the full URL of the document

Variable: variable1 - specify the variable name to store the result

Examples:

Get Dublin Core - Subject XMP metadata property from Current Item: Encoded Absolute URL and store in Variable:XMPSubject

9.8 Is Image Compressed (HarePoint)

If image at this URL uses compression

This workflow condition analyzes image file content to check wherher the image is compressed or is not. In some cases, e.g. in case of TIFF images, it is impossible to determine if the image is compressed or is not only by file extension.

Parameters:

this URL - Full URL of the picture in Document Library.

Examples:

If image at Current Item:Encoded Absoulte URL is supported

If image at Current Item:Encoded Absoulte URL uses compression

Log Conversion is not required to the workflow history list

else

Convert image at Current Item:Encoded Absoulte URL to JPG format and store URL of converted image in Variable: ConvertedImageUrl. Overwrite existing image.

then Delete item at Current Item:Encoded Absoulte URL list item url.

This workflow automatically converts uncompressed images to JPG format.

9.9 Is Image Type Supported (HarePoint)

If image at this URL is supported

This workflow condition analyzes file extension and returns true if the file type is supported by the Product image processing engine.

File formats supported by the Product image processing engine:

- BMP files [reading, writing]
- DDS files [reading]
- EXR files [reading, writing]
- Raw Fax G3 files [reading]
- GIF files [reading, writing]
- HDR files [reading, writing]
- ICO files [reading, writing]
- IFF files [reading]
- JNG files [reading]
- JPEG/JIF files [reading, writing]
- JPEG-2000 File Format [reading, writing]
- JPEG-2000 codestream [reading, writing]
- KOALA files [reading]
- Kodak PhotoCD files [reading]
- MNG files [reading]
- PCX files [reading]
- PBM/PGM/PPM files [reading, writing]
- PFM files [reading, writing]
- PNG files [reading, writing]
- Macintosh PICT files [reading]
- Photoshop PSD files [reading]
- RAW camera files [reading] (see note below)
- Sun RAS files [reading]
- SGI files [reading]
- TARGA files [reading, writing]

- TIFF files [reading, writing]
- WBMP files [reading, writing]
- XBM files [reading]
- XPM files [reading, writing]

Note: Product can load RAW files from digital photo cameras (CRW/CR2, NEF, RAF, DNG, MOS, KDC, DCR, etc), virtually all RAW formats are supported).

Parameters:

this URL - Full URL of the picture in Document Library.

Examples:

If image at Current Item:Encoded Absoulte URL is supported

If image at Current Item:Encoded Absoulte URL uses compression

Log Conversion is not required to the workflow history list

else

Convert image at Current Item:Encoded Absoulte URL to JPGformat and store URL of converted image in Variable: ConvertedImageUrl. Overwrite existing image.

then Delete item at Current Item:Encoded Absoulte URL list item url.

This workflow automatically converts uncompressed images to JPG format.

9.10 Resize Image

Resize image at specified url. Set width: 0, height: 0 (keep aspect ratio: yes) and append '_resize' to filename. Do not process image if it is smaller than target.

This workflow action is used to resize image. With this action you can easily avoid huge images from modern digital cameras in SharePoint galleries.

Note: See 'Get Image Size' activity overview for the list of supported file formats.

Parameters:

specified url - Full URL of the picture in Document Library.
width 0 - Maximum width of converted picture, in pixels.

height 0 - Maximum height of converted picture, in pixels.

keep aspect ratio - Yes or no. In the first case, the output picture will have one dimension exactly as specified by width/height parameters, and another dimension may be smaller than the one specified by width/height.

append '_resize' to filename - To options are available: save output file with the new name with prefix, or to overwrite source file.

Do not process - Process or not process source file if it has smaller dimensions than the one specified by height and width parameters.

Examples:

If Current Item: File Type equals jpg

Resize image at Current Item: Encoded Absolute URL. Set width: 800, height: 600 (keep aspect ratio: yes) and overwrite existing file. Do not process image if it is smaller than target.

9.11 Resize Images in Folder

Resize images in folder at this url. Set width: 0, height: 0 (keep aspect ratio: yes). Store resized images in folder at specified url and append '_resize' to filename. Do not process image if it is smaller than target. Do not delete source images

This workflow action is used to resize all images of supported formats at Document Library or folder of Document Library.

Note: See 'Get Image Size' activity overview for the list of supported file formats.

Parameters:

this url - URL of source Document Library or folder.

width 0 - Maximum width of converted picture, in pixels.

height 0 - Maximum height of converted picture, in pixels.

keep aspect ratio - Yes or no. In the first case, the output picture will have one dimension exactly as specified by width/height parameters, and another dimension may be smaller than the one specified by width/height.

specifed url - URL of the target Document Library or folder. Source and target Document Libraries may be the same.

append '_resize' to filename - To options are available: save output file with the new name with prefix, or to overwrite source file in case the source and the target Document Libraries are the same.

Do not process - Process ordo not process, or just copy source file to the target folder if it has smaller dimensions than specified by height and width parameters.

Do not delete - Delete or not delete source images.

9.12 Resize Images in Library

Resize images in folder at this library. Set width: 0, height: 0 (keep aspect ratio: yes). Store resized images in specified library and append '_resize' to filename. Do not process image if it is smaller than target. Do not delete source images

This workflow action is used to resize all images of supported formats at Document Library or folder of Document Library.

Note: See 'Get Image Size' activity overview for the list of supported file formats.

Parameters:

this library - Document Library with source images.

width 0 - Maximum width of converted picture, in pixels.

height 0 - Maximum height of converted picture, in pixels.

keep aspect ratio - Yes or no. In the first case, the output picture will have one dimension exactly as specified by width/height parameters one, and another dimension may be smaller than the specified by width/height.

specifed library - Target Document Library for converted images. Source and target Document Libraries may be the same.

append '_resize' to filename - To options are available: save output file with the new name with prefix, or to overwrite source file in case the source and the target Document Libraries are the same.

Do not process - Process or not process, or just copy source file to the target folder if it has smaller dimensions than the specified by height and width parameters.

Do not delete - Delete or do not delete source images.

9.13 Rotate Image

Rotate image at specified url 90 degrees Clockwise.

This workflow action is used to rotate the image in one from supported formats to 90, 180 or 270 degrees clockwise or counter-clockwise.

Note: See 'Get Image Size' activity overview for the list of supported file formats.

Parameters:

specified url - URL of the image in Document Library.
90 - How many degrees to rotate the image: 90, 180, 270.
Clockwise - How to rotate the image: clockwise or counter-clockwise.

9.14 Set EXIF Property on Image

Set EXIF property Title on image at this url to variable value

This workflow action sets the specified EXIF property for the specified image.

Parameters:

Title - the required EXIF property to be set can be selected from the drop-down list.
Available values: Title, Subject, Artist, Copyright, Date and Time, Software, GPS Latitude, GPS Longtitude, GPS Altitude
this url - the URL of the image
variable - the value of EXIF property to be set

Examples:

Set EXIF property Title on image at Current Item:Encoded Absolute URL to Trafalgar Square value

9.15 Set Transparent Color

Set #FF00FF color of the image at specified url transparent, convert to PNG.

This workflow action is used to set transparent color for an image in one of supported formats. Image with the transparency can be saved in PNG or GIF formats only.

The transparent color is a property of the image, transparency is not a property of the color. Once you have set a color to be the transparent color, any regions of the image in that color that were drawn previously will be transparent.

Note: See 'Get Image Size' activity overview for the list of supported file formats.

Parameters:

#FF00FF - Color to mark as transparent. You can use hexadecimal RGB values (like #RRGGBB) or the name of system-defined color from this list: http://msdn.microsoft.com/en-us/library/system.drawing.knowncolor(VS.90).aspx
specified url - URL of the image in Document Library.
PNG - Output format for the image with transparency: PNG or GIF.

9.16 Set XMP Property

Set specified MPX metadata property in document at this url to this value, trigger events

This workflow action sets the specified XMP metadata property for specified document to specified value

Parameters:

specified MPX metadata - choose XMP metadata property from the drop-down list.
document at this url - specify the full url of the document
this value - specify the value to be set.
In case if property field contains multiple properties, they should be delimited by semicolon (;)
Example:
Set Dublin Core - Subject XMP metadata property in url to construction; house; building
trigger events - select to trigger/do not trigger item change event.

Examples:

Set Dublin Core - Description XMP metadata property in Current Item: Encodede Absolute URL to Some description here, trigger events.

10. Lists activities

10.1 Add Content Type to List

Add this content type to list at this list url.

This workflow action adds specified content type to the list.

Parameters:

this - Name of the content type.

this list url - Full URL of the target list.

10.2 Add Existing Column to Content Type

Add this existing column to specified content type on the site at current site url

This workflow action adds and existing column to a content type.

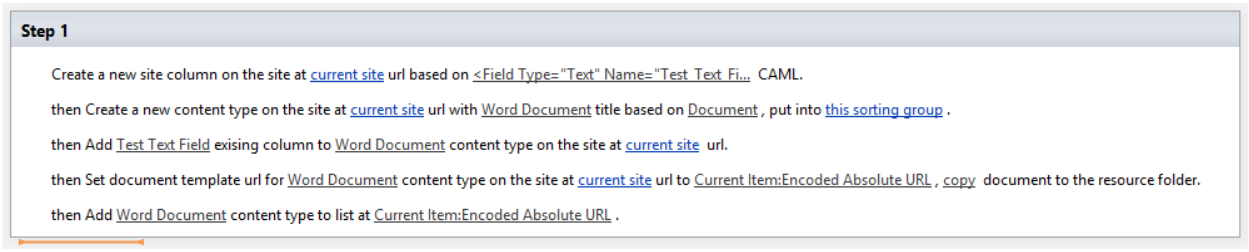
Parameters:

this existing column - specify the name of the existing column to be added

specified content type - specify the content type name to which the column will be added

current site - (optional) specify the URL of the site

Examples:


Step 1
Create a new site column on the site at current site url based on <Field Type="Text" Name="Test_Text_Fi... CAML.
then Create a new content type on the site at current site url with Word Document title based on Document , put into this sorting group .
then Add Test Text Field exising column to Word Document content type on the site at current site url.
then Set document template url for Word Document content type on the site at current site url to Current Item:Encoded Absolute URL , copy document to the resource folder.
then Add Word Document content type to list at Current Item:Encoded Absolute URL .

CAML: <Field Type="Text" Name="Test_Text_Field" DisplayName="Test Text Field" />

10.3 Approve Item at Url

Approve **this list item url** item with **specified** comment.

This workflow action is used to approve list item in specified by URL.

Note: To get list item URL you can use “Get List Item Url” action.

Parameters:

this list item url - Full URL of list item.
specified - Comment text.

Examples:

Get Display Form url for list item Current Item and store result in Variable: F1

Get Edit Form url for list item Current Item and store result in Variable: F2

Get Native Form url for list item Current Item and store result in Variable: F3

*then Log [%Variable: F1%] * [%Variable: F2%] * [%Variable: F3%] to the workflow history list*

then Approve [%Variable: F3%] item with Approved by workflow comment

Result workflow history list:

http://gorlach7/Shared Documents/Forms/DispForm.aspx ID=214 * http://gorlach7/Shared Documents/Forms/EditForm.aspx ID=214 * http://gorlach7/Shared Documents/2144750538 Doc064.pdf

10.4 Build String from List

Build string using this template and delimiter from list items at this list url by this CAML query. Store result in Variable: variable.

This workflow action is used to build string variable using values of specified fields of multiple list items.

Parameters:

this template - String containing one or more macros in format <%FieldName%>. Where FieldName is a display name or internal name of list column.
For example: *Item <%Title%> was created at <%Created%>.*
delimiter - Optional. Delimiter between strings built from template for each list item. By default - new line (line feed).

this list url - Full URL of SharePoint list.
this CAML query - String with the CAML query.
Variable: variable - Output string variable.

Examples:

Build string using Item <%Title%> was created at <%Creat... and delimiter from list items at http://sharepoint/Lists/Tasks/ by <Where> <IsNotNull> <Fi... . Store results in Variable: variable.

the Log Variable: variable to the workflow history list

10.5 Calculate Aggregation Function

Calculate SUM on this field from all items recursively in this list and store result in Variable:variable1

This workflow action is used to calculate SUM, COUNT, AVG, MIN or MAX aggregation function on list items field values. Note that empty field values are skipped by all these functions including COUNT.

Aggregation functions and supported field formats:

- SUM, calculate summarized value: Number, Integer, String.
- COUNT, count non-empty items: Any format.
- AVG, calculate average value: Number, Integer.
- MIN, seeking the minimum value: Number, Integer, Date/Time, String.
- MAX, seeking the maximum value: Number, Integer, Date/Time, String.

The third parameter of this action contains multi-line string with list items filter in one from four available formats:

1. Default empty string means 'all items'.

2. List of GUIDs of list items, each line contains one GUID (value of GUID property of list item).
3. List of case-insensitive masks (with * and wildcards) for names of list items or list items names (value of Name property of list item). Each line contains one mask or name.
4. Filter in the CAML (Collaborative Application Markup Language) Query format, see below.

CAML format is a bit complicated, but very flexible. It is based on XML. The example of correct CAML filter, it is a body of CAML Query element:

```
<Where>
    <And>
      <Eq>
        <FieldRef Name='FirstName' />
        <Value Type='Text'>Alexey</Value>
      </Eq>
      <Contains>
        <FieldRef Name='Email' />
        <Value Type='Text'>@gmail.com</Value>
      </Contains>
    </And>
  </Where>
```

Note, that you should use internal names of fields in the CALM Query.

For the CAML reference, see:

- Writing CAML queries: http://sharepointmagazine.net/articles/writing-caml-queries-for-retrieving-list-items-from-a-sharepoint-list

- CAML Query Schema: http://msdn.microsoft.com/en-us/library/ms467521.aspx
- CAML Query Element: http://msdn.microsoft.com/en-us/library/ms471093.aspx
- Freeware tool to build CAML Query: http://www.u2u.be/Software

Parameters:

SUM - Aggregation function. One from following options: *SUM, COUNT, AVG, MIN, MAX*.
this field - Field name.
all - Items filter. All items by default, or the list of items names, or the list of list items GUIDs, or the CAML query.
recursively - count inside summary tasks which may also include child items.
Or select **non-recursively** to skip summary task chlid items.
this list - specify the SharePoint list.
Variable:variable1 - Output variable to store result.
Note: The type of output variable should match with the type of result. E.g. attempt to store Date/Time result to Number variable will cause an exception.

10.6 Copy InfoPath Attachments to Document Library

Copy Info Path attachment specified by XPATH of list item at this url to document library at specified url. Overwrite existing items. Store copied item url in Variable: variable.

This workflow action is used to copy InfoPath attachments of specified list item to document library.

Parameters:

specified by XPATH - XPath query string. See examples section for details.
this url - Full URL of list item.
specified url - Full URL of target document library.

Overwrite - If the document library alrady contains file with the same name, you can choose to *Overwrite* it or to skip this file (*Do not overwrite*).

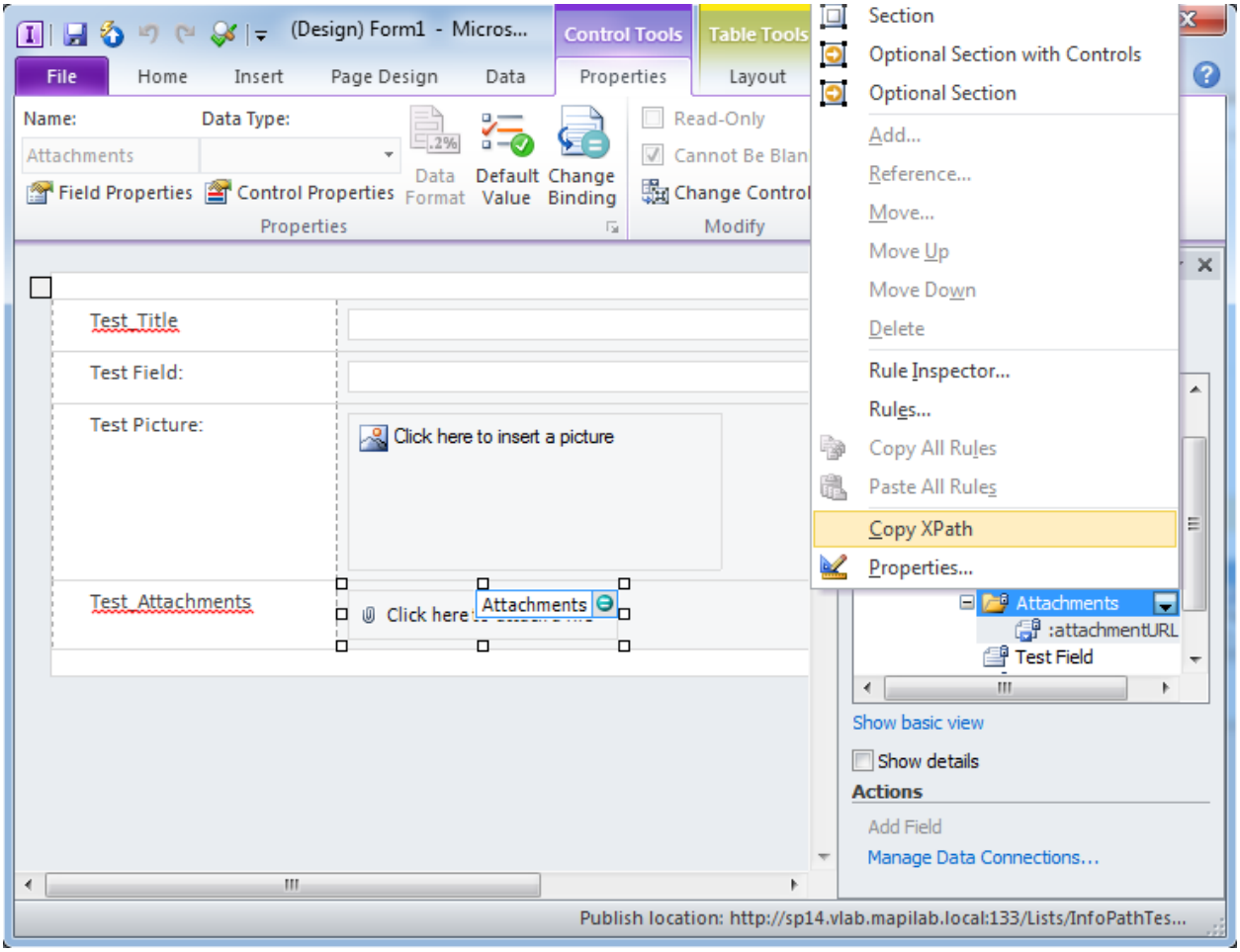
Variable: variable - Full URL of the new attachment.

Examples:

Copy Info Path attachment /dfs:myFields/dfs:dataFields/my:SharePointListItem_RW/my:Attachments of list item at Variable: listItemUrl to document library at Variable: docLibUrl. Overwrite existing items. Store copied item url in Variable: variable.

You can combine multiple XPath queries using vertical bar ('|') sign, e.g. /my:myFields/my:MoreAttachments | /my:myFields/my:AttachFile.

You can retrieve XPath string using Microsoft InfoPath Designer:



10.7 Copy List Item at Url

Copy list item at this url to the list at this list url. Store output item url in Variable: variable1. Trigger events.

This workflow action is used to copy list item from one list to another.

Note: This action can copy multiple list items, if an array of URLs is provided as the input parameter.

Parameters:

this url - Full URL of the source list item. Arrays of the URLs are also supported; in this case all items for this array will be copied.

this list url - Full URL of the target list.

Variable: variable1 - Output string variable for the URL of copied list item.

Trigger - Specifies whether to trigger an item create event.

10.8 Copy List Item Attachments to Document Library

Copy attachments of list item at this url to document library at specified url. Overwrite if items exist.

This workflow action is used to copy attachments from specified list item to document library specified by URL.

See Also: "Copy List Item Attachments to Document Library (advanced)" - this action is identical, but allows you to save the URLs of the saved attachments in an array variable.

See Also: "Upload Files to List Item Attachments" action.

Parameters:

this url - Full URL of list item.

specified url - Full URL of target document library.

Overwrite - If the document library already contains file with the same name, you can choose to *Overwrite* it, to skip this file (*Do not overwrite)*, or to *Append unique ID* to the file name.

10.9 Copy List Item Attachments to Document Library (advanced)

Copy attachments of list item at this url to document library at specified url. Overwrite if items exist. Store urls of output items in Variable:UrlsArray

This workflow action is used to copy attachments from specified list item to document library specified by URL, and save the URLs of the output items to an array.

See Also: "Copy List Item Attachments to Document Library" in case if you don't need to have the output URLs saved to a variable.
See Also: "Upload Files to List Item Attachments" action.

Parameters:

this url - Full URL of list item.

specified url - Full URL of target document library.

Overwrite - If the document library already contains file with the same name, you can choose to *Overwrite* it, to skip this file (*Do not overwrite)*, or to *Append unique ID* to the file name.

Variable:UrlsArray - the array variable to save the URLs of the items that have been copied to target library.

10.10 Copy List Item Attachments to List Item

Copy attachments of list item at this url to list item at specified url. Overwrite existing items.

This workflow action is used to copy attachments of one list item to attachments of another list item.

See Also: "Upload Files to List Item Attachments" action.

Parameters:

this url - Full URL of source list item.

specified url - Full URL of target list item.

Overwrite - If target item already contains attachment with the same name, then you can choose one from two options. *Overwrite* - to replace file by new one. *Do not overwrite* - to skip this file.

10.11 Copy List Item to Different Farm

Copy list item at this url to the list at this list url. Store output item url in Variable: variable1.

This workflow action is used to copy list item from one list to another located on a different SharePoint farm.

Important Note: the action *'Set Target Site Options'* always should be used before this action!

Additionally you can use '*Set Encrypted Passwords*' action and *Encrypt Credentials* tool to specify the credentials in secure way.

See Also: 'Set Encrypted Passwords' and 'Set Target Site Options' actions.

Parameters:

this url - Full URL of the source list item.
this list url - Full URL of the target list located on a different SharePoint farm.
Variable: variable1 - Output string variable for the URL of copied list item.

10.12 Create Content Type

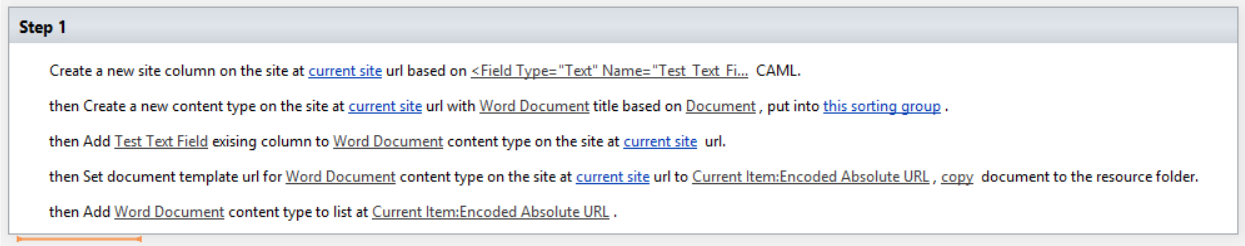
Create a new content type on the site at current site url with this title based on this content type, put into this sorting group.

This workflow action creates a new content type.

Parameters:

current site - (optional) specify the URL of the site where the new content type needs to be created.
this title - the title of the new content type
based on this content type - the name of the parent content type
this sorting group - (optional) the name of the group where the new content type should be placed.

Examples:


Step 1
Create a new site column on the site at current site url based on <Field Type="Text" Name="Test_Text_Fi... CAML.
then Create a new content type on the site at current site url with Word Document title based on Document , put into this sorting group .
then Add Test Text Field exising column to Word Document content type on the site at current site url.
then Set document template url for Word Document content type on the site at current site url to Current Item:Encoded Absolute URL , copy document to the resource folder.
then Add Word Document content type to list at Current Item:Encoded Absolute URL .

CAML: <Field Type="Text" Name="Test_Text_Field" DisplayName="Test Text Field" />

10.13 Create List

Create list on the current site with this title and this description based on template with this name, show on Quick Launch: yes . Store list url to Variable: variable.

Creates new SharePoint list or document library using specified template.

Can also create lists on other sites (not only on the current site).

Parameters:

on the current site - (Optional) Specify the URL of the site where the list needs to be created. Leave *current site* to create a list on a current site.
this title - Title of new list.
this description - Description of the new list.
this name - Name of the list of document library template.
show on Quick Launch: yes - specify whether to display this list on Quick Launch or not.
Variable: variable - This variable will contain the full URL of the created list or document library

10.14 Create List Item at Url

Create list item with specified values at this list url. Store new list item url in Variable:variable1. Trigger events.

This workflow action is used to create new list item and set the values of list item fields. Values are specified as multi-line string of one from two available formats.

First, the format in style of INI file. For example:

Position=Team Leader

First Name=John

Last Name=Smith

Age=23

Second, is XML format, like:

<item>

<field title="Position">Team Leader</Field>

<field title="First Name">John</Field>

<field title="Last Name">Smith</Field>

<field title="Age">23</Field>

</item>

See Also: "Update List Item at Url" action.

Parameters:

specified values - Values for list item fields in INI or XML format.
For **Hyperlink** type fields it is also possible to specify description using *Url#description*: **Example:** *Title=new task* *Url=http://www.harepoint.com* *Url#description=HarePoint website* **Managed metadata** columns are supported, it is possible to specify term label names from Terms Set. Multiple labels should be delimited by a semicolon (;)

Example:
FieldName=Label1;Label2;Label3

this list url - Full URL of list.

Variable: variable1 - Output string variable for the URL of created list item.

Trigger - Specifies whether to trigger an item create event.

10.15 Create List Item on Different Farm

Create list item with specified values at this list url. Store new list item url in Variable: variable.

This workflow action is used to create new list item at the different SharePoint farm and set the values of list item fields. Values are specified as multi-line string of one from two available formats.

First, the format in style of INI file. For example:

```
Position=Team Leader
First Name=John
Last Name=Smith
Age=23
```

Second, is XML format, like:

```
<item>
<field title="Position">Team Leader</Field>
<field title="First Name">John</Field>
<field title="Last Name">Smith</Field>
<field title="Age">23</Field>
</item>
```

Important Note: the action *'Set Target Site Options'* always should be used before this action!

Additionally you can use '*Set Encrypted Passwords*' action and *Encrypt Credentials* tool to specify the credentials in secure way.

See Also: 'Set Encrypted Passwords' and 'Set Target Site Options' actions.

Parameters:

specified values - Values for list item fields in INI or XML format.
this list url - Full URL of list on different farm.
Variable: variable - Output string variable for the URL of created list item.

Examples:

Example 1:

Use site at http://tragetsite/one url as target site. Logon using COMPANY\someuser username and *somepassword* password.

then Create list item with Somevalue=127 at http://targetsite/one/Lists/Orders ...

Example 2:

Set encrypted passwords with BjhjgHUuj+jhucu... string

then Use site at http://targetsite/one url as target site. Logon using this user username and this password password.

then Create list item with Somevalue=127 at http://targetsite/one/Lists/Orders ...

10.16 Create Site Column

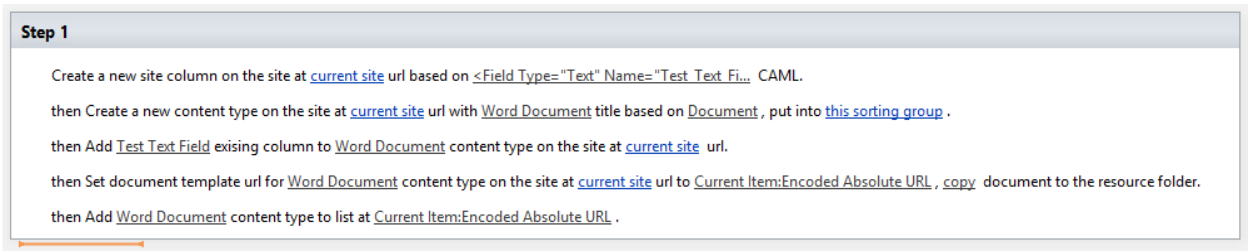
Create a new site column on the site at current site url based on this CAML.

This workflow action creates a new site column on the site based on CAML query.

Parameters:

Current site - (optional) Specify the URL of the site where the column needs to be created.
this CAML - CAML query.
For details on CAML syntax please refer to http://msdn.microsoft.com/en-us/library/office/ms437580(v=office.15).aspx

Examples:


Step 1
Create a new site column on the site at current site url based on <Field Type="Text" Name="Test_Text_Fi... CAML.
then Create a new content type on the site at current site url with Word Document title based on Document , put into this sorting group .
then Add Test Text Field exising column to Word Document content type on the site at current site url.
then Set document template url for Word Document content type on the site at current site url to Current Item:Encoded Absolute URL , copy document to the resource folder.
then Add Word Document content type to list at Current Item:Encoded Absolute URL .

CAML: <Field Type="Text" Name="Test_Text_Field" DisplayName="Test Text Field" />

10.17 Delete All List Items

Delete all list items in current list. Store number of deleted items in Variable: variable.

This workflow action is used to delete list items of current list.

The first parameter of this action contains multi-line string with list items filter in one from four available formats:

1. Default empty string means 'all items'.
2. List of GUIDs of list items, each line contains one GUID (value of GUID property of list item).
3. List of case-insensitive masks (with * and wildcards) for names of list items or list items names (value of Name property of list item). Each line contains one mask or name.
4. Filter in the CAML (Collaborative Application Markup Language) Query format, see below.

CAML format is a bit complicated, but very flexible. It is based on XML. The example of correct CAML filter, it is a body of CAML Query element:

```
<Where>
  <And>
    <Eq>
      <FieldRef Name='FirstName' />
```

```
      <Value Type='Text'>Alexey</Value>
    </Eq>
    <Contains>
      <FieldRef Name='Email' />
      <Value Type='Text'>@gmail.com</Value>
    </Contains>
  </And>
 </Where>
```

Note, that you should use internal names of fields in the CALM Query.

For the CAML reference, see:

- Writing CAML queries: http://sharepointmagazine.net/articles/writing-caml-queries-for-retrieving-list-items-from-a-sharepoint-list
- CAML Query Schema: http://msdn.microsoft.com/en-us/library/ms467521.aspx
- CAML Query Element: http://msdn.microsoft.com/en-us/library/ms471093.aspx
- Freeware tool to build CAML Query: http://www.u2u.be/Software/

Parameters:

all - Items filter. All items by default, or the list of items names, or the list of list items GUIDs, or the CAML query.
current list - (Optional) Specify the URL of the list where the items should be deleted. By default, current list is used.
Variable: variable - Output integer variable to store number of deleted items.

10.18 Delete Item at Url

Delete item at specified list item url.

This workflow action is used to delete list item specified by URL.

Note: To get list item URL you can use “Get List Item Url” action.

Parameters:

specified - Full URL of list item.
Note: the array of full URLs of list items is also supported.

10.19 Delete Item on Different Farm

Delete item at specified list item url on different farm.

This workflow action is used to delete list item at different SharePoint farm.

Important Note: the action '*Set Target Site Options*' always should be used before this action!

Additionally you can use '*Set Encrypted Passwords*' action and Encrypt Credentials tool to specify the credentials in secure way.

See Also: '*Set Encrypted Passwords*' and 'Set Target Site Options' actions.

Parameters:

specified - Full URL of list item at different SharePoint farm.

10.20 Download InfoPath Attachments to Folder

Download Info Path attachment specified by XPATH of list item at this url to this folder. Overwrite existing items.

This workflow action is used to save InfoPath attachments of specified list item to local disk or to network share.

Note: In order to specify file access credentials, add the "Set Network File Access" activity to the same workflow step first.

Parameters:

specified by XPATH - XPath query string. See "Copy InfoPath Attachments to Document Library" action examples section for details.
this url - Full URL of list item.
this folder - Local disk or network path.
Overwrite - If the document library alrady contains file with the same name, you can choose to *Overwrite* it or to skip this file (*Do not overwrite*).

10.21 Find List Item at Url

Find item in the list at this url where item's field equals value. Store top 1000 result urls in Variable: array.

This workflow action is used to find list items by the value of the specified field.

The list of supported operators and types:

Operator	String	Integer	Date/Time
equals	+, case insensitive	+	+
not equals	+, case insensitive	+	+
is greater than	-	+	+
is greater than or equals to	-	+	+
is less than	-	+	+
is less than or equals to	-	+	+
begins with	+, case insensitive	-	-
ends with	+, case insensitive	-	-
contains	+, case insensitive	-	-
doesn't contain	+, case insensitive	-	-
is empty	+	+	+
is not empty	+	+	+
is in array	+, case insensitive	+	-

Parameters:

this url - Full URL of list
field - The name of the field to search for.
equals - Comparison opeator.

value - Value to search for.
top 1000 **-** specify number of results to be stored
Variable: array - Output array variable with URLs of list items.

Examples:

Find item in the list at Variable: listUrl where item's Company equals MAPILab Ltd. Store result urls in Variable: array.

10.22 Find List Item by CAML

Find item in the list at this url by this CAML query. Store top 1000 result urls in Variable: array.

This workflow action is used to find list items using CAML query.

See "Calculate Aggregation Function" activity description for examples and references on CAML queries.

See Also: "Find List Item at Url" activity.

Parameters:

this url - Full URL of list.
this CAML query - String with the CAML query.
top 1000 - Specify number of results to be stored.
Variable: array - Output array variable with URLs of list items.

Examples:

Find item in the list at Variable: listUrl by <Where><And><Eq><FieldRef.... Store result urls in Variable: array.

10.23 Find List Item by CAML on Different Farm

Find item in the list at this url by this CAML query. Store top 1000 result urls in Variable: array.

This workflow action is used to find list items at the different SharePoint farm using CAML query.

See "Calculate Aggregation Function" activity description for examples and references on CAML queries.

Important Note: the action '*Set Target Site Options*' always should be used before this action!

Additionally you can use '*Set Encrypted Passwords*' action and Encrypt Credentials tool to specify the credentials in secure way.

See Also: '*Set Encrypted Passwords*' and 'Set Target Site Options' actions.

See Also: For the complete list of supported operators and types see 'Find List Item at Url' action.

Parameters:

this url - Full URL of list at the different SharePoint farm.
this CAML query - String with the CAML query.
top 1000 - specify number of results to be stored
Variable: array - Output array variable with URLs of list items.

10.24 Find List Item on Different Farm

Find item in the list at this url where item's field equals value. Store top 1000 result urls in Variable: array.

This workflow action is used to find list items on different SharePoint farm by the value of the specified field.

Important Note: the action '*Set Target Site Options*' always should be used before this action!

Additionally you can use '*Set Encrypted Passwords*' action and Encrypt Credentials tool to specify the credentials in secure way.

See Also: '*Set Encrypted Passwords*' and 'Set Target Site Options' actions.

See Also: For the complete list of supported operators and types see 'Find List Item at Url' action.

Parameters:

this url - Full URL of list at the different SharePoint farm.
field - The name of the field to search for.

equals - Comparison opeator.

value - Value to search for.

top 1000 - specify number of results to be stored

Variable: array - Output array variable with URLs of list items.

10.25 Get Field Value at Url

Get value from this field of list item at this url (current version) and store result in Variable: variable.

This workflow action is used to get a field value from list item specified by URL.

Parameters:

this field - Field name.

this url - Full URL of list item.

current version - specify the version of the list item.

If the versioning is enabled for the list, you can get the data from the previous versions of the item.

Two formats can be used:

- -1, -2, -3, ... - to specify previous versions relatively to the current version
- 1.0, 2.0, 3.0 - to specify the required version number directly

Variable: variable - Output variable for the field value.

10.26 Get List Item Attachment Content

Get content of the attachment with this name at this list item url and store in Variable: variable.

This workflow action is used to retrieve contents of specfied list item attachment into output variable.

Parameters:

with this name - Name of the attachement.

this list item url - Full URL of list item.

Variable: variable - Output variable.

Examples:

Get content of the attachment OriginalMessage.eml at Current Item: Encoded Absolute URL and store in Variable: eml.

then Get sender email from Variable: eml body and store in Variable: Sender Address.

then Get reply to from Variable: eml body and store in Variable: Reply To.

the Update list item with Sender Address=[%Variable: Sender Add... at Current Item: Encoded Absolute URL list item url. Do not trigger events.

10.27 Get List Item Attachments

Get Names of attachments of list item at this url and store to Variable: ArrayVariable.

This workflow action is used to get names, file sizes or URLs of list item attachments.

See Also: "Join Text" and "Get Array Items Count" activities.

Parameters:

Names - One from three options: *Names*, *URLs* or *File Sizes*.

this - Full URL of list item.

Variable: ArrayVariable - Output string array with names or URLs, or number array with file sizes.

Examples:

Get Names of attachments of list item at Current Item:Encoded Absolute URL url and store to Variable: namesArray.

then Get the number of items in Variable: namesArray and store in Variable: attachmentsCount.

then Log Variable: attachmentsCount to the workflow history list

10.28 Get List Item Url

Get Display Form url for list item this list item (current version) and store result in Variable:variable.

This workflow action is used to get URL of specified list item, or URL of specified list item edit form or URL of specified list item view form.

Parameters:

Display Form - One from three options: *Display Form, Edit Form, Native Form*. In the case of documents library, *Native Form* is the URL of file.
this list item - List item, e.g. Current Item.
current version - specify the version of the list item.
If the versioning is enabled for the list, you can get the data from the previous versions of the item.
Two formats can be used:
• -1, -2, -3, ... - to specify previous versions relatively to the current version
• 1.0, 2.0, 3.0 - to specify the required version number directly
Variable:variable- Output string variable.

Examples:

Get Display Form url for list item Current Item (current version) and store result in Variable: F1

Get Edit Form url for list item Current Item (current version) and store result in Variable: F2

Get Native Form url for list item Current Item (current version) and store result in Variable: F3

*then Log [%Variable: F1%] * [%Variable: F2%] * [%Variable: F3%] to the workflow history list*

then Approve [%Variable: F3%] item with Approved by workflow comment

Result workflow history list:

http://gorlach7/Shared Documents/Forms/DispForm.aspx ID=214 * http://gorlach7/Shared Documents/Forms/EditForm.aspx ID=214 * http://gorlach7/Shared Documents/2144750538 Doc064.pdf

10.29 Get List Url

Get Display Form url of this list and store to Variable: variable.

This workflow action is used to get one from seven possible URL types for a list on current SharePoint site:

- Display Form
- Edit Form
- New Form
- Native Form
- Mobile Display Form
- Mobile Edit Form
- Mobile New Form

Note: Most workflow actions are works with Native Form of URL.

Parameters:

Display Form - Type of URL to return.
this - List on current site.
Variable: variable - Output string variable.

10.30 Get Workflow Task Property

Get this property value from workflow task at this url and store result in variable

This workflow action allows to get values of extended properties of a workflow task.

Parameters:

this property - the property name. Can be left blank, in this case it will log the names of all available properties.
this url - specify the URL of the workflow task item
variable - specify the variable to store the property value.
If propery is not found, *Null* will be stored.

10.31 If List Exists (HarePoint)

If list with this name exists on the site at this url

This workflow condition is used to determine if the specified list exists on an indicated site

Parameters:

list with this name - specify the List name
at this url - specify the site URL where to look for the required List.

10.32 Is Current Item Checked Out (HarePoint)

If current item is checked out

This workflow condition returns true if the current item is checked out.

10.33 Is Specified Item Checked Out (HarePoint)

If item at specified url is checked out

This workflow condition returns true if the item specified by URL is checked out.

Parameters:

specified - Full URL of list item.

10.34 Reject Item at Url

Reject list item at this url item with specified comment.

This workflow action is used to reject items in specified list.

Note: To get list item URL you can use “Get List Item Url” action.

Parameters:

this - Full URL of list item.
specified - Comment text.

10.35 Remove All Attachments

Remove all attachments from list item at this url. Store number of removed attachments to Variable: variable.

This workflow action is used to remove attachments by name or by file mask from the list item specified by URL.

Parameters:

all - Multi-line string with one attachment file name or file mask at line.

this url - Full URL of list item.

Variable: variable - Output integer variable to store number of removed attachments.

10.36 Remove Content Type from List

Remove this content type from list at this list url.

This workflow action removes specified content type from the list.

Parameters:

this - Name of the content type.

this list url - Full URL of the target list.

10.37 Remove InfoPath Attachments

Remove Info Path attachment specified by XPATH of list item at this url.

This workflow action is used to remove InfoPath attachments from the list item specified by URL.

Parameters:

specified by XPATH - XPath query string. See "Copy InfoPath Attachments to Document Library" action examples section for details.

this url - Full URL of list item.

10.38 Set Content Type Document Template

Set document template url for this content type on the site at current site url to this url, copy document to the resource folder.

This workflow action sets a document template for a content type.

Parameters:

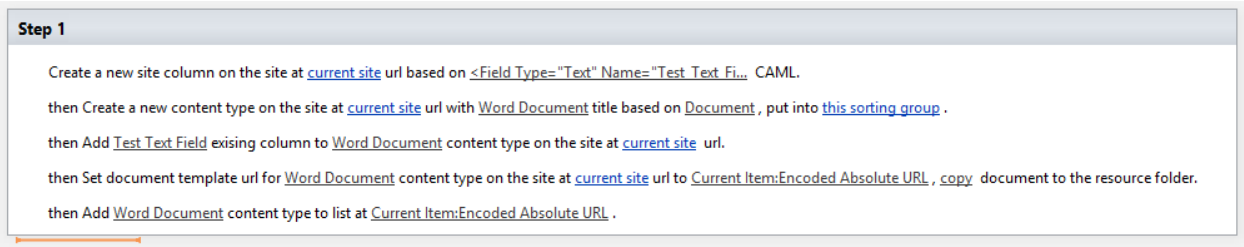
this content type - specify the content type name

current site - (optional) specify the URL of the site

to this url - specify the URL of the document that will be used as a template

copy - select to copy/do not copy the document template to the resource folder.

Examples:


Step 1
Create a new site column on the site at current site url based on <Field Type="Text" Name="Test_Text_Fi... CAML.
then Create a new content type on the site at current site url with Word Document title based on Document , put into this sorting group .
then Add Test Text Field exising column to Word Document content type on the site at current site url.
then Set document template url for Word Document content type on the site at current site url to Current Item:Encoded Absolute URL , copy document to the resource folder.
then Add Word Document content type to list at Current Item:Encoded Absolute URL .

CAML: <Field Type="Text" Name="Test_Text_Field" DisplayName="Test Text Field" />

10.39 Set Default Content Type on List

Set this content type as default content type on the list at this list url.

This workflow action sets specified content type as default for the specified list. If list does not contains specified content type, then it will be added first.

Parameters:

this content type - Name of the content type.

this list url - Full URL of the target list.

10.40 Update List Item at Url

Update list item with specified values at this list item url. Trigger events.

This workflow action is used to set values of list item fields. It is possible to update **multiple** list items in a single action by providing an **array** of URLs as an input parameter.

Values are specified as multi-line string of one from two formats.

First, the format in style of INI file. For example:

```
Position=Team Leader
First Name=John
Last Name=Smith
Age=23
```

Second, is XML format, like:

```
<item>
<field title="Position">Team Leader</field>
<field title="First Name">John</field>
<field title="Last Name">Smith</field>
<field title="Age">23</field>
</item>
```

See Also: "Create List Item" action.

Parameters:

specified values - Values for list item fields in INI or XML format.

For **Hyperlink** type fields it is also possible to update description using *Url#description*:

Example:
Title=new task
Url=http://www.harepoint.com
Url#description=HarePoint website

Mutiselect Lookup fields are supported, values can be either comma (,) or semicolon (;) delimited.

Managed metadata columns are supported, it is possible to specify term label names from Terms Set. Multiple labels should be delimited by a semicolon (;)

Example:
FieldName=Label1;Label2;Label3

this - Full URL of list item. **Arrays** of full URLs are also supported, in this case **multiple** items will be updated.

Trigger - Specifies whether to trigger an item update event.

Examples:

Example 1: Updating single item

Update list item with Title=[%Current Item:Title%] (updated) at Current Item:Encoded Absolute URL list item url. Trigger events.

Example 2: Updating multiple items

Find item in the list at http://sitename.local/Lists/Tasks where item's Title equals test. Store top 100 result urls in Variable: array.

then Update list item with Title=test (updated) at Variable: array list item url. Trigger events.

10.41 Update List Item on Different Farm

Update list item with specified values at this list item url.

This workflow action is used to set values of list item fields on different SharePoint farm. Values are specified as multi-line string of one from two formats.

First, the format in style of INI file. For example:

Position=Team Leader

First Name=John

Last Name=Smith

Age=23

Second, is XML format, like:

<item>

<field title="Position">Team Leader</Field>

<field title="First Name">John</Field>

<field title="Last Name">Smith</Field>

<field title="Age">23</Field>

</item>

Important Note: the action '*Set Target Site Options*' always should be used before this action!

Additionally you can use '*Set Encrypted Passwords*' action and Encrypt Credentials tool to specify the credentials in secure way.

See Also: '*Set Encrypted Passwords*' and 'Set Target Site Options' actions.

Parameters:

specified values - Values for list item fields in INI or XML format.
this - Full URL of list item at different sharpoint farm.

10.42 Update Workflow Task at Url

Update wokflow task with specified values at this url

This workflow action allows to set the values of the workflow task properties, including extended properties.

One of the scenarios how this action can be used, is to complete a workflow task from a workflow:

http://blogs.msdn.com/b/malag/archive/2009/05/20/how-to-programmatically-complete-approval-workflow-task.aspx

Parameters:

specified values - specify values workflow task properties in the following format: <PropertyName1>=<value1>
<PropertyName2>=<value2>
...
this url - specify the URL of the workflow task item

10.43 Upload Files to List Item Attachments

Upload files from those paths to list item at this url. Overwrite existing items. Store number of uploaded files to Variable: variable.

This workflow action is used to upload files from local disk, network share, document library or web server to list item attachments.

Note: In order to specify file access credentials, add the "Set Network File Access" activity to the same workflow step first.

Parameters:

those paths - Multi-line string where each line contains a path with file name or URL. Local disk or network share paths can contain masks in file name (note, that recursive search in subfolders is not supported).

this url - Full URL of list item.

Overwrite - What to do with existing attachments with the same name. *Not overwrite* means skip existing attachments.

Variable: variable - Output integer variable with the number of uploaded files.

10.44 Wait for Field Change in Current Item Until Date

Wait for field to equal value until date this time. Store timeout flag to Variable: isTimeOut

This workflow action is waiting till field value satisfies the specified logical expression. If the specified date has reached, time out is triggered.

Parameters:

field - specify the field to be checked

to equal - logical operand. Available values: To equal / To not equal.

value - specify the target value

this time - specify the date when timeout will occur

Variable: isTimeOut - 'false' when no timeout occured, and 'true' when timeout has occured

10.45 Wait for Field Change in Current Item With Timeout

Wait for field to equal value with timeout 0 days, 0 hours, 5 minutes. Store timeout flag to Variable: isTimeOut

This workflow action is waiting till field value satisfies the specified logical expression. If the specified time limit has been exceeded, time out is triggered.

Parameters:

field - specify the field to be checked
to equal - logical operand. Available values: To equal / To not equal
value - specify the target value
0 days - days for timeout
0 hours - hours for timeout
5 minutes - minutes for timeout
Variable: isTimeOut - 'false' when no timeout occured, and 'true' when timeout has occured

11. Network activities

11.1 Create Directory on FTP Server

Create specified directory in this folder on FTP server.

This workflow action is used to create directory on FTP server.

To specify server name, login and password place "Set FTP Server Options" activity the current workflow step first.

Parameters:

specified - Directory name on FTP server.
this - Folder name to create.

Examples:

See example in "Set FTP Server Options" activity description.

11.2 Delete Directory from FTP Server

Delete this directory from FTP server. Do not delete directory if directory is not empty.

This workflow actions is used to delete specified directory on the FTP server.

To specify server name, login and password place "Set FTP Server Options" activity the current workflow step first.

Parameters:

this - Directory name to delete.

Do not delete directory - Specifies whether to delete child items or not.

11.3 Delete File from FTP Server

Delete this file from this directory on FTP server.

This workflow action is used to delete specified file from the FTP server.

To specify server name, login and password place "Set FTP Server Options" activity the current workflow step first.

Parameters:

this file - File name to delete.

this directory - Directory name on the FTP server.

11.4 Fetch RSS Feed to Html

Fetch RSS from this url print by [DATE]: [TITLE] [DESCRIPTION] template and store in Variable: OutputHtml

This workflow action is used to fetch all items from RSS feed and print it by specified template.

Parameters:

this url - URL of RSS feed.

template - Printing template for RSS item. [DATE], [URL], [TITLE] and [DESCRIPTION] in template will be replaced by corresponding RSS item fields.

OutputHtml - Output string variable, will contains concatenated string with all feed items.

Examples:

Fetch RSS from http://www.zdnet.com/search t=7&mode=rss&tag=mantle_skin;content print by [DATE]: [TITLE] [DESCRIPTION] template and store in Variable: html

then Create document from <html><body>[%Variable: html%]</body></html> in http://site/Shared Documents/news.html. Overwrite existing document.

11.5 Fetch RSS Item

Fetch RSS from this url and store top item number 1's title in Variable: Title, link in Variable: Link, description Variable: Description and publication date in Variable: Date.

This workflow action is used to fetch top item from RSS feed.

Parameters:

this url - RSS feed address.

item number **1**'s - Number of item to fetch.

Variable: Title - Output string variable for RSS item title.

Variable: Link - Output string variable for RSS item link.

Variable: Description - Output string variable for RSS item text.

Variable: Date - Output date variable for RSS item publication date.

11.6 Send HTTP GET Request

Send HTTP request to this url using GET method with this headers. Store response in Variable: variable1 and status code in Variable: variable2.

This workflow action is used to send HTTP GET request.

Note: In order to specify HTTP request credentials, add the "*Set HTTP Request Options*" activity to the same workflow step first.

Parameters:

this url - Target URL with parameters.
this headers - Additional request headers each on new line. For example: *Accept-Language: fr*
Variable: variable1 - Output string variable to receive remote server response, usually contains requested web page body.
Variable: variable2 - Output integer variable for HTTP status code (example: 200).

Examples:

Encode [%Current Item:Approver Comments%] using url encoder and store in Variable: param.

then Send HTTP request to http://mysite/ q=[%Variable: param%] using GET method with this headers. Store response in Variable: pageBody and status code in Variable: code.

11.7 Send HTTP OAuth Request

Send HTTP OAuth request to this url using GET method with these headers and this request. Store response in variable.

This workflow action is used to send HTTP OAuth request. Various HTTP methods available.

Note: Set OAuth Authentication action needs to be used prior this action.

See also: **Set OAuth Authentication** action.

Parameters:

this url - this is the API endpoint, the URL of the web-resource to send the request to.
Please refer to "Developers API" documentation for details.
using GET method - select the HTTP method. Possible values:

- GET
- POST
- PUT
- DELETE

Please refer to "Developers API" documentation to find out which method should be used.

these headers - (Optional) specify the additional headers for this request, if necessary.
this request - (Optional) specify the request body, if necessary.
Variable - specify the variable where the server response will be saved to.

Examples:

How to work with xero.com: Creating a new contact

(see also **Set OAuth Authentication** action)

Step 1. Obtain the base64-encoded private key. It can be generated using the command line utility *openssl*:

openssl gensra -out privatekey.pem 1024

more details here: http://developer.xero.com/documentation/advanced-docs/public-private-keypair/

Step 2. Obtain consumer key and consumer secret here: https://app.xero.com/Application/List

Step 3. Set up a workflow (this is the example to **create a new Contact**):

- Set xero.com OAuth authentication. Use LLWHHD7WS505 key, PRNCI5KNCHU76NSP secret, sign request with RSA-SHA1 MIICWwIBlIjchPkmJKwcOL..., request token at this url, get access token at this url
- then Send HTTP OAuth request to https://api.xero.com/api.xro/2.0/cont... using PUT method with these headers and <Contact> <Name>ABCDEFG</Name> <F... request. Store response in Variable:variable
- then Log Variable:variable to the workflow history list

Comments for workflow action 1:

- The consumer key and consumer secret from **Step 2** are used in this action in respective parameters
- The base64-encoded private key from **Step 1** is used in the additional parameter after RSA-SHA1

Comments for workflow action 2:

According to the xero.com API documentation, the parameters are filled as follows:

- request URL is https://api.xero.com/api.xro/2.0/Contacts
- HTTP method that is used is PUT

- request body is:

```
<Contact>
 <Name>ABCDEFG</Name>
 <FirstName>John</FirstName>
 <LastName>Smith</LastName>
 <EmailAddress>john.smith@gmail.com</EmailAddress>
 <Addresses>
  <Address>
   <AddressType>POBOX</AddressType>
    <AddressLine1>P O Box 123</AddressLine1>
    <City>Wellington</City>
    <PostalCode>6011</PostalCode>
   </Address>
 </Addresses>
 <BankAccountDetails>01-0123-0123456-00</BankAccountDetails>
 <TaxNumber>12-345-678</TaxNumber>
 <AccountsReceivableTaxType>OUTPUT</AccountsReceivableTaxType>
 <AccountsPayableTaxType>INPUT</AccountsPayableTaxType>
 <DefaultCurrency>NZD</DefaultCurrency>
</Contact>
```

11.8 Send HTTP POST Request

Send HTTP request to this url using POST method with this headers and this request. Store response in Variable: variable1 and status code in Variable: variable2.

This workflow action is used to send HTTP POST request.

Note: In order to specify HTTP request credentials, add the "*Set HTTP Request Options*" activity to the same workflow step first.

Parameters:

this url - Target URL.
this headers - Additional request headers each on new line. For example: *Accept-Language: fr*
this request - The body of the request. For example: *name=john&age=23*

Variable: variable1 - Output string variable to receive remote server response, usually contains requested web page body.
Variable: variable2 - Output integer variable for HTTP status code (example: 200).

Examples:

Encode [%Current Item:Approver Comments%] using url encoder and store in Variable: param.

then Send HTTP request to http://mysite/script.aspx using POST method with this headers and query=[%Variable: param%] request. Store response in Variable: pageBody and status code in Variable: code.

11.9 Send XMPP Instant Message

Send XMPP instant Message to this JID by using JID: this JID, password: this password and optional server: this server.

This workflow action is used to send instant message to XMPP-compatible messaging clients like Google Talk.

List of compatible clients is available here: http://xmpp.org/xmpp-software/clients/

Parameters:

Message - Message text
to **this JID** - Receiver's JID
using JID: **this JID** - Sender's JID
this password - Sender's password
this server - Network host to logging on

Examples:

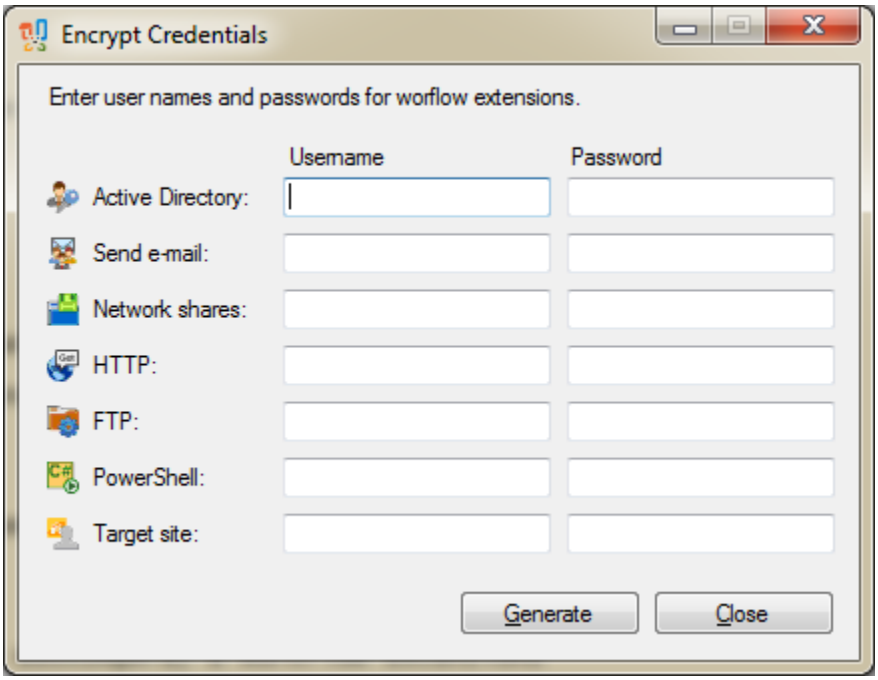
Send XMPP message Hello! to myfriend@gmail.com by using JID: myaccount@gmail.com, password: mypassword and optional server: talk.google.com.

11.10 Set Encrypted Passwords

Set encrypted passwords with this string.

This workflow action is used to setup authorization settings in encrypted form. This action can be used instead of any other action that sets the usernames and passwords.

To generate encrypted string for this action use "Encrypt Credentials" tool:



Parameters:

this - Base64 string generated by the "Encrypt Credentials" tool.

11.11 Set Exchange Web Service Options

Use Exchange Server at this url or e-mail address and this user and this password to authenticate against web service, Do not convert message body to plain text, use SharePoint computer time zone, impersonate as current user.

This workflow action is setting Exchange Web Service options which are requred to use 'Send Meeting Request' action

Parameters:

this url or e-mail address - specify either URL of the EWS (e.g. https://exch.example.org/EWS/Exchange.asmx), or e-mail address of the user to determine this URL using Autodiscover Service (this option is triggered when '@' symbol is contained in the string).

this user - user name for authentication (should be left blank if credentials are provided via Encrypt Credentials tool).
Note: the meeting request will be sent on behalf of this account, unless *impersonate as current user* parameter is selected in this workflow action - see below for details.
this password - the password for authentication (should be left blank if credentials are provided via Encrypt Credentials tool).
Note: You can use the Encrypt Credentials tool to encrypt the password, in this case specify the Username and Password in Web Service field and use 'Set encrypted password' action.
Do not convert message body - select to convert message body to plain text or not.
Use **SharePoint computer** time - (optional) specify the time zone for the meeting request. By default SharePoint Computer time zone will be used.
If necessary, time zone can be specified explicitly. The available names for time zones can be viewed in the following system registry key:
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Time Zones
impersonate as current user - it is possbile to impersonate as currenet user, so the meeting request will be sent on behalf of the user who launched the workflow.
Two conditions should be met for successful impersonation:
1. The user who will launch the workflow need to have a **valid e-mail address** in his profile on the Exchange server being used.
2. The user account (who will launch the workflow) need to have ***Send As*** **permission**. Please refer to http://msdn.microsoft.com/en-us/library/office/bb204095(v=exchg.140).aspx for more details.

Examples:

Use Exchange Server at https://exch.example.org/EWS/Exchange.asmx and this user and this password to authenticate against web service, Do not convert message body to plain text, use SharePoint computer time zone, impersonate as current user

11.12 Set FTP Server Options

Use specified FTP server, disable SSL, login this user and this password.

This workflow action is used to set FTP server name, login, password and other options for other FTP-related activities like "Upload Document to FTP Server" or "Create Directory on FTP Server".

This activity must be placed above all other FTP-related activities in the step. You can use this activity several times in one workflow step to use different FTP servers.

Note: The password will be stored as a plain text in the workflow XOML file and may be accessed by the URL like*http://sharepoint/Workflows/workflowname/workflowname.xoml* or by SharePoint Designer.

Parameters:

specified - FTP server name or IP-address. By default port 21 is used, you can specify port with colon, like ftp.company.com:121.
disable - Disable or enable SSL. Note, that FTP servers with SSL usually listen for incoming connections at port 990.
login - FTP server login. If empty, *Anonymous* user name is used.
password - FTP server password. May be empty.

Examples:

Use ftp.company.com:990 FTP server, enable SSL, login root user and hU2BnkM13 password.

then Create mydir directory in web/temp folder on FTP server.

then Upload Current Item document to FTP server into web/temp/mydir folder. If file already exists upload with unique name.

then Use 10.1.1.1 FTP server, disable SSL, login this user and this password.

then Upload Tasks list item attachments to FTP server into incoming folder. If the file already exists do nothing.

In the example above, in the last activity, when choosing Tasks list item, specify searching 'Attachments=Yes' in the "Choose List Item" window. Unfortunatelly, SharePoint designer does not visualize searching options in workflow window.

11.13 Set HTTP Request Options

Use this user and this password to authenticate against website.

This workflow action is used to set HTTP login and password for other HTTP-related activities like "Send HTTP GET Request".

Note: The password will be stored as a plain text in the workflow XOML file and may be accessed by the URL like *http://sharepoint/Workflows/workflowname/workflowname.xoml* or by SharePoint Designer.

Parameters:

this user - User name that will be used for HTTP authentication.
this password - Password that will be used for HTTP authentication.

11.14 Set OAuth Authentication

Set xero.com OAuth authentication. Use this consumer key, this consumer secret, sign request with default, request token at this url, get access token at this url

This workflow action is used to set OAuth authentication.

See also: **Send HTTP OAuth Request** action

Parameters:

xero.com - select the site to authenticate against. Possible values:
• xero.com
• general
this consumer key - specify the consumer key used for authentication. You can obtain it from the respective web site, usually in "Developers API" section.
this consumer secret - specify the consumer secret used for authentication. You can obtain it from the respective web site, usually in "Developers API" section.
sign request with default - select the method to to sign the requests to the server. Possible values:
• default
• plain text
• hmac-sha1
• rsa-sha1 (**required for xero.com**)
Please refer to "Developers API" section for more details.
When **rsa-sha1** is selected, an **additional parameter** would need to be filled:

RSA key - paste here the base64-encoded private key from **.pem** file. (The instructions how to obtain .pem file are available in the description on the respective site).

request token at this url - (Optional) specify the URL to obtain the request token. It is used during the authentication process, e.g. for digital signature.

access token at this url - (Optional) specify the URL to exchange request token with access token. It is used to access API functionality.

Examples:

See **Send HTTP OAuth Request** action for an example on how to use these actions for **xero.com**.

11.15 Set Web Service Options

Use default web service base url and this username and this password

This workflow action is setting Web Service options which are reqired to use 'Get Cell Value' and 'Set Cell Value'

Parameters:

default web service - (optional) you can specify URL for Excel Web Services in case if it's not detected properly with the default setting.

Note: Site address URL should be used

this username - specify the username

this password - specify the password

Note: You can use the Encrypt Credentials tool to encrypt the password, in this case specify the Username and Password in Web Service field and use 'Set encrypted password' action.

11.16 Shorten Url

Shorten this url with goo.gl using this login and this password and store result in Variable:variable

This workflow action generates the short URL using online providers for short URLs.

Parameters:

this url - specify URL to be shorted

with goo.gl - select the provider

Currently two provides available:

- Goo.gl - does not require registration, so login and password are not required
- Bit.ly - requires registration on Bitly.com, login and password should be specified in action parameters.

this login - specify the login name (only for bit.ly)

this password - specify the password (only for bit.ly)

store result in Variable:variable - the shortend URL is stored in this variable.

11.17 Upload Document to FTP Server

Upload specified document to FTP server into this folder. If file already exists do nothing.

This workflow action is used to upload list item on FTP server.

To specify server name, login and password place "Set FTP Server Options" activity to current workflow step first.

Parameters:

specified - List item to upload.

this - FTP directory name.

do nothing - What to do if the file already exists. Four options are available: *do nothing*, *treat as an error* - generate error and stop workflow, *upload with unique name* - add unique identifier like {11fba226-59d6-4f31-b9c4-e43ddc67e048} to the end of file name, or *overwrite* existing file.

Examples:

See example in "Set FTP Server Options" activity description.

11.18 Upload List Item Attachments to FTP Server

Upload specified list item attachments to FTP server into this folder. If the file already exists do nothing.

This workflow action is used to upload list item attachments to FTP server directory.

To specify server name, login and password place "Set FTP Server Options" activity to current workflow step first.

Parameters:

specified - List item whose attachments will be uploaded.
this - Directory name on FTP server.
do nothing - What to do if the file already exists. Four options are available: *do nothing*, *treat as an error* - generate error and stop workflow, *upload with unique name* - add unique identifier like {11fba226-59d6-4f31-b9c4-e43ddc67e048} to the end of file name, or *overwrite* existing file.

Examples:

See example in "Set FTP Server Options" activity description.

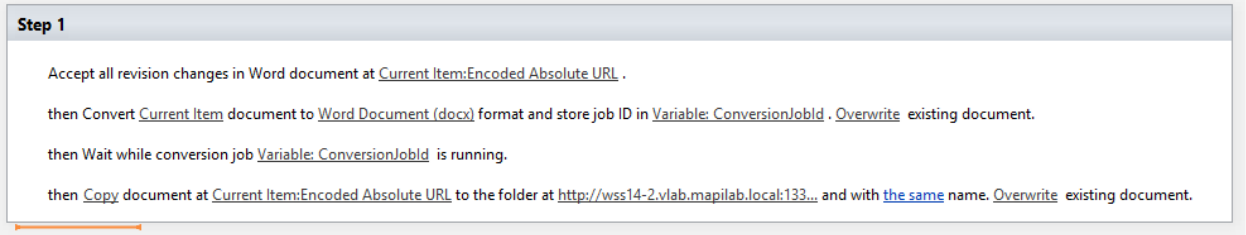
12. Office activities

12.1 Accept Revision Changes in Word Document

Accept all revision changes in Word document at this url

This workflow action accepts all revision changes in the specified Word document.

Note: To update the fields in the document, conversion *docx to docx* is required. Please refer to the following screenshot for proper usage of this action:


Step 1
Accept all revision changes in Word document at Current Item:Encoded Absolute URL .
then Convert Current Item document to Word Document (docx) format and store job ID in Variable: ConversionJobId . Overwrite existing document.
then Wait while conversion job Variable: ConversionJobId is running.
then Copy document at Current Item:Encoded Absolute URL to the folder at http://wss14-2.vlab.mapilab.local:133... and with the same name. Overwrite existing document.

Parameters:

this url - the URL of the Word document

Examples:

Accept all revision changes in Word document at Current Item:Encoded Absolute URL

12.2 Add New Row into Excel Document

Add new row from this array into specified sheet in this document. Start looking for the last row from this cell.

This workflow action adds a new row to Excel document. It can automatically detect the nearest empty row to paste the values.

Note: Only .xlsx files are supported.

Parameters:

this array - the array of values that will be added. The array can be created using "Create Array" or "Split Text" actions.

specified sheet - the sheet name in Excel document.

this document - the full URL to Excel file.

this cell - if this cell is empty, the values from the array will be added to this cell and following cells in the same row.

If this cell is not empty, this workflow action will first check next cells in the same column if they are empty. As soon as empty cell is found, the values from the array will be added to this cell and following cells in the same row.

12.3 Add New Row into Word Document

Add new row into this repeating section content control of this .docx document

This workflow action adds a copy of the last row in the Repeating Section Content Control. After this has been added it can be modified using 'Update Word Document' action.

To create Repeating Section Content Control, highlight the necessary row in Word Document, and on the ribbon on Developer Tab click 'Add Repeating Section Content Control' button.

Parameters:

this repeating section - the name of the repeating content control

this .docx document - the URL of the word document to be modified

Examples:

Add new row into Repeating content control of Current Item document

An article with detailed example can be found here:
http://www.harepoint.com/Products/HarePointWorkflowExtensions/Example-Repeating-Content-Workflow.aspx

12.4 Find Text in Word Document

Find this text in this .docx document, match case: no. Store result in Variable: variable.

This workflow action is used to check presence of specified substring at Microsoft Word document.

In this version, only DOCX documents are supported.

Note: Word Automation Services are not required for this workflow action.

Parameters:

this text - Input string to find in Word document.
this - Reference to document, e.g. Current Item.
no - Is the search case-sensitive.
Variable: variable - Output boolean variable.

12.5 Get Cell Value

Get value of this cell of specified sheet in this document and store in Variable: CellValue.

This workflow action is used to get cell value at Microsoft Excel document at SharePoint server.

Note: This workflow action requires Excel Services installed on the server.

Parameters:

this - Cell address, e.g. A1.
specified - Name of the sheet, e.g. Sheet1.
this - Reference to XLS or XLSX document, e.g. Current Item.

Variable: CellValue - Output string variable.

12.6 Get Office Document Property

Get this property of this office document and store in Variable: variable

This workflow action is used to get the property of the document at SharePoint server. With this action, you can get the following properties:

- Title
- Author
- Keywords
- Subject
- Comments
- Last Modified By
- Category
- Manager
- Company

In the current version, only Open Office XML (Microsoft Office 2007-2013) formats are supported: DOCX, XLSX, PPTX.

See Also: Open Office XML at Wikipedia (http://en.wikipedia.org/wiki/Office_Open_XML).

Note: Word or Excel Automation Services are not required for this workflow action.

Parameters:

this property - Office document property name.

this - Reference to document, e.g. Current Item.
Variable: variable - Output string variable.

12.7 Read Word Document

Read this content control of this .docx document and store its value in Variable: variable

This workflow action is used to get text of named content control of Microsoft Word document. See 'Update Word Document' activity overview to find out how to place named content control to Microsoft Word document.

In this version, only DOCX documents are supported.

Note: Word Automation Services are not required for this workflow action.

Parameters:

this content control - Content control name, see overview section for details.
this - Reference to document, e.g. Current Item.
Variable: variable - Output string variable.

12.8 Read Word Document To Text

Read this .docx document and store its text in Variable: variable

This workflow action is used to get text of Microsoft Word document to string variable.

In this version, only DOCX documents are supported.

Note: Word Automation Services are not required for this workflow action.

Parameters:

this - Reference to document, e.g. Current Item.
Variable: variable - Output string variable.

12.9 Replace Text in Word Document

Replace all occurencies of this string in this document with this text, match case: no.

This workflow action is used to replace text in Microsoft Word document. This action can be used for generation of bills and invoices, filling forms and addresses on envelopes within workflow.

In this version, only DOCX documents are supported.

See Also: 'Update Word Document' action.

Note: Word Automation Services are not required for this workflow action.

Parameters:

all occurencies - With this action it is possible to replace *all occurences*, *first occurence* or *last occurence* of specified pattern.
this string - String to replace.
this - Reference to document, e.g. Current Item.
this text - New string.
no - Case sensitive: *yes* or *no*.

12.10 Set Cell Value

Set value of this cell of specified sheet in this document to this value.

This workflow action is used to set cell value at Microsoft Excel document at SharePoint server.

Note: This workflow action requires Excel Services installed on the server.

Parameters:

this - Cell address, e.g. A1.
specified - Name of the sheet, e.g. Sheet1.
this - Reference to XLS or XLSX document, e.g. Current Item.
this value - String, Integer, Number, Date/Time or Boolean variable with new cell value.

12.11 Set Office Document Property

Set this property of this office document to this value

This workflow action is used to set the value of document property at SharePoint server. With this action, you can set the value for the following properties:

- Title
- Author
- Keywords
- Subject
- Comments
- Last Modified By
- Category
- Manager
- Company

In the current version, Open Office XML (Microsoft Office 2007-2013) formats are supported only: DOCX, XLSX, PPTX.

See Also: Open Office XML at Wikipedia (http://en.wikipedia.org/wiki/Office_Open_XML).

Note: Word or Excel Automation Services are not required for this workflow action.

Parameters:

this property - Office document property name.
this - Reference to document, e.g. Current Item.
this value - String variable with new value.

12.12 Update Picture in Word Document

Set this picture content control of this document to picture at this url

This workflow action sets the picture located at specified URL to picture content control of the specified document.

See Also: 'Update Word Document' action.

Parameters:

this picture content control - the picture content control to be set
this document - the URL of the document to be modified
this url - the direct URL to the picture within this SharePoint farm

Examples:

Set logo of Invoices document to picture at https://spsite/Shared%20Documents/newpic.jpg

12.13 Update Word Document

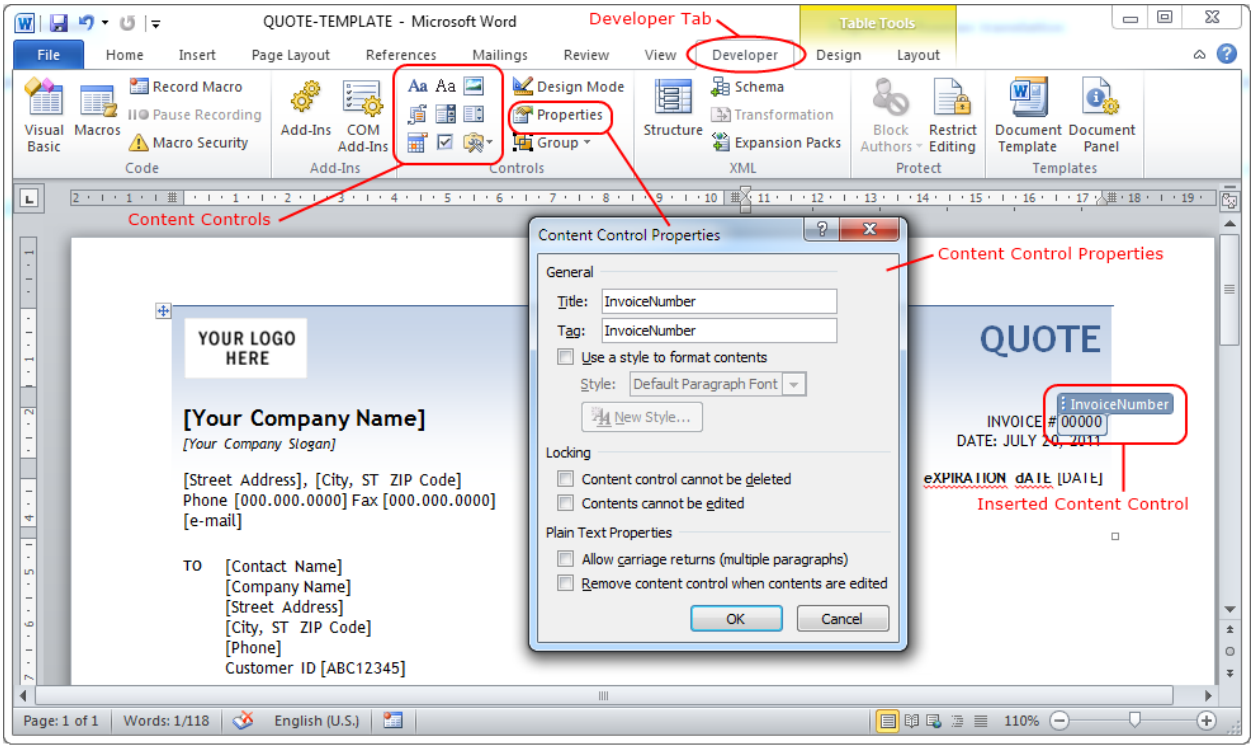
Set value of this content control of this .docx document to this value, text format: plain text

This workflow action is used to update text of named content control in Microsoft Word document. This action can be used for generation of bills and invoices, filling forms and addresses on envelopes within workflow.

To place content control, you should open Developer Tab in Microsoft Word:

1. Click the Microsoft Office Button , and then click Word Options.
2. Click Popular.
3. Select the Show Developer tab in the Ribbon check box, and then click OK.

On Developer Tab, you should select content control on the Ribbon, insert it into the document and assign name using Properties button:



.

See Also: 'Replace Text in Word Document' action.

Note: Word Automation Services are not required for this workflow action.

Parameters:

this content control - Content control name, see overview section for details.
this .docx document - Reference to document, e.g. Current Item.
this value - String with the new text for content control.
plain text - specify the format of the text. Available values: *plain text, HTML*.

Examples:

Set InvoiceNumber of Current Item .docx document to 12345, text format:HTML

This workflow replace content of "InvoiceNumber" content control with 12345. See overview sections for instructions how to place named content control to the Microsoft Word document.

13. Permissions activities

13.1 Change Permissions Inheritance at Url

Start inheriting permissions at this url

This workflow action is used to start or stop inheriting permissions for SharePoint site, list, list folder or list item specified by URL.

Note: May need to place this action into Impersonation Step. Otherwise, action will be executed under context of the user who started the workflow.

Parameters:

Start/stop - Action to do with inheritance.
this url - Full URL of a SharePoint site, list, list folder or list item.

Examples:

Set Variable: item to [%Workflow Context:Current Item URL%].

then Stop inheriting permissions at Variable: item.

13.2 Change Permissions Inheritance for List

Start inheriting permissions for this list

This workflow action is used to start or stop inheriting of list's permissions.

Note: May need to place this action into Impersonation Step. Otherwise, action will be executed under context of the user who started the workflow.

Parameters:

Start/stop - Action to do with inheritance.
this list - Target list.

13.3 Change Permissions Inheritance for List Item

Start inheriting permissions for this list item

This workflow action is used to start or stop inheriting of list's item permissions.

Note: May need to place this action into Impersonation Step. Otherwise, action will be executed under context of the user who started the workflow.

Parameters:

Start/stop - Action to do with inheritance.
this list item - Target list item.

13.4 Delete All Permissions at Url

Delete all permissions at this url

This workflow action is used to delete all permissions from SharePoint site, list, list folder or list item specified by URL.

Note: May need to place this action into Impersonation Step. Otherwise, action will be executed under context of the user who started the workflow.

Parameters:

this url - Full URL of a SharePoint site, list, list folder or list item.

13.5 Delete All Permissions from List

Delete all permissions from this list

This workflow action is used to delete all permissions from list.

Note: May need to place this action into Impersonation Step. Otherwise, action will be executed under context of the user who started the workflow.

Parameters:

this list - Target list.

13.6 Delete All Permissions from List Item

Delete all permissions from this list item

This workflow action is used to delete all permissions from list item.

Note: May need to place this action into Impersonation Step. Otherwise, action will be executed under context of the user who started the workflow.

Parameters:

this list item - Target list item.

13.7 Delete All User Permissions at Url

Delete all permissions at this url for this user.

This workflow action is used to delete all permissions for specified user from SharePoint site, list, list folder or list item specified by URL.

Note: May need to place this action into Impersonation Step. Otherwise, action will be executed under context of the user who started the workflow.

Parameters:

this url - Full URL of a SharePoint site, list, list folder or list item.
this user - All permissions for this user or group will be removed.

13.8 Delete All User Permissions from List

Delete all permissions from this list for this user

This workflow action is used to delete all permissions for specified user from list.

Note: May need to place this action into Impersonation Step. Otherwise, action will be executed under context of the user who started the workflow.

Parameters:

this list - Target list.
this user - All permissions for this user or group will be removed.

13.9 Delete All User Permissions from List Item

Delete all permissions from this list item for this user

This workflow action is used to delete all permissions for specified user from list item.

Note: May need to place this action into Impersonation Step. Otherwise, action will be executed under context of the user who started the workflow.

Parameters:

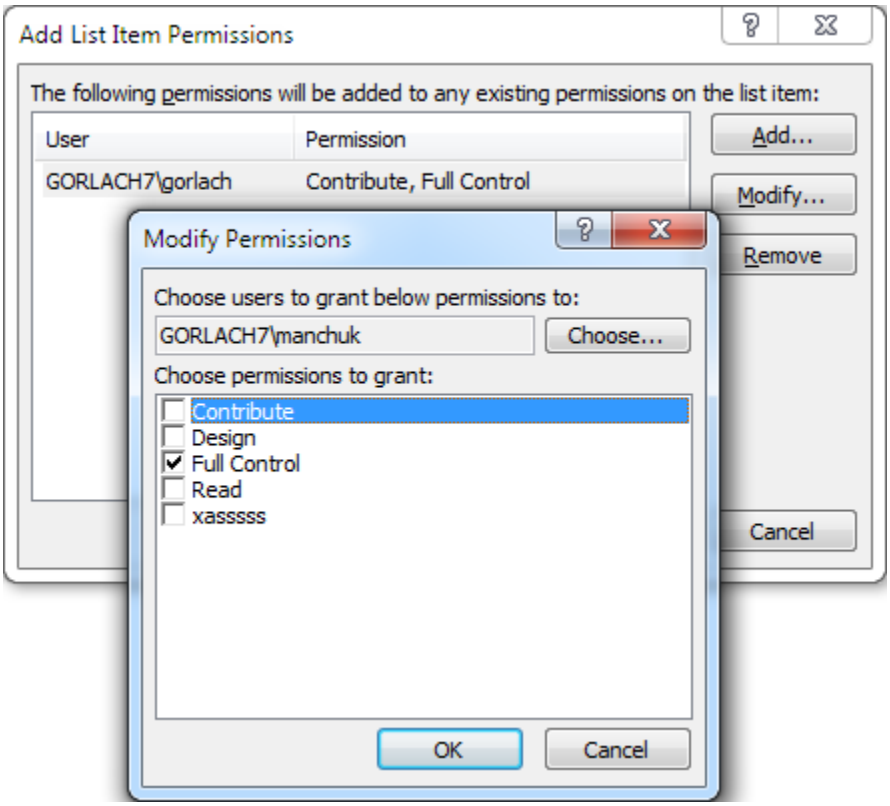
this list item - Target list item.

this user - All permissions for this user or group will be removed.

13.10 Delete Permissions at Url

Delete specified permissions at this url

This workflow actions is used to remove site users permissions from SharePoint site, list, list folder or list item specified by URL. In the Remove List Item Permissions window you can choose multiple users and select any permissions for them including Out Of The Box and custom permissions.



Note: May need to place this action into Impersonation Step. Otherwise, action will be executed under context of the user who started the workflow.

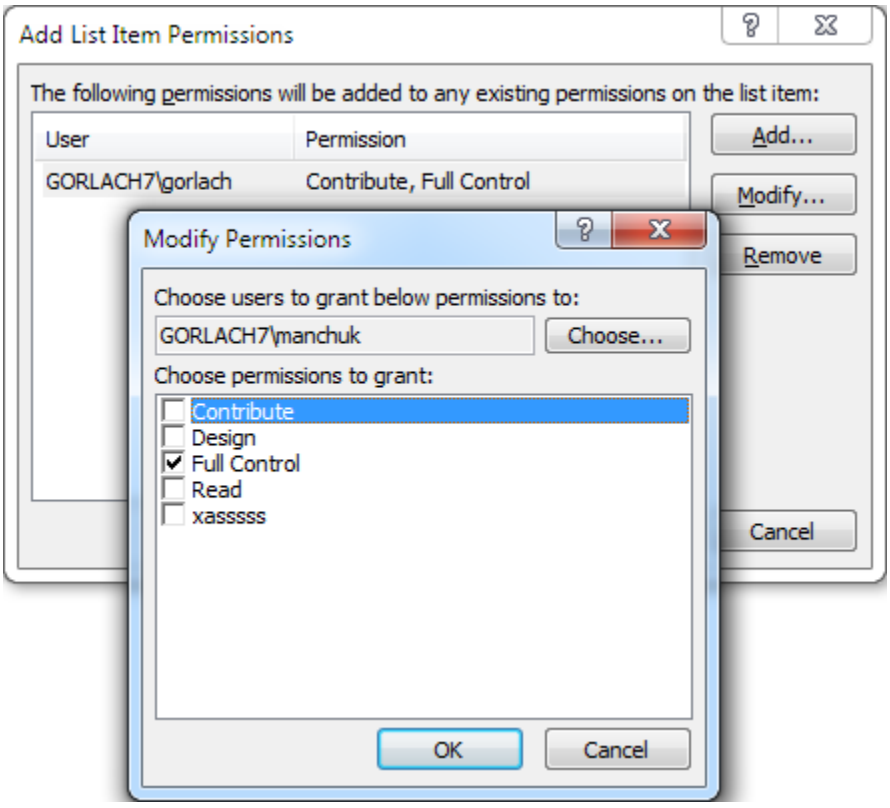
Parameters:

specified - Special-type array of the users and they permissions.
this url - Full URL of a SharePoint site, list, list folder or list item.

13.11 Delete Permissions from List

Delete specified permissions from this list

This workflow actions is used to remove permissions to site users for specified list. In the Remove List Item Permissions window you can choose multiple users and select any permissions for them including Out Of The Box and custom permissions.



Note: May need to place this action into Impersonation Step. Otherwise, action will be executed under context of the user who started the workflow.

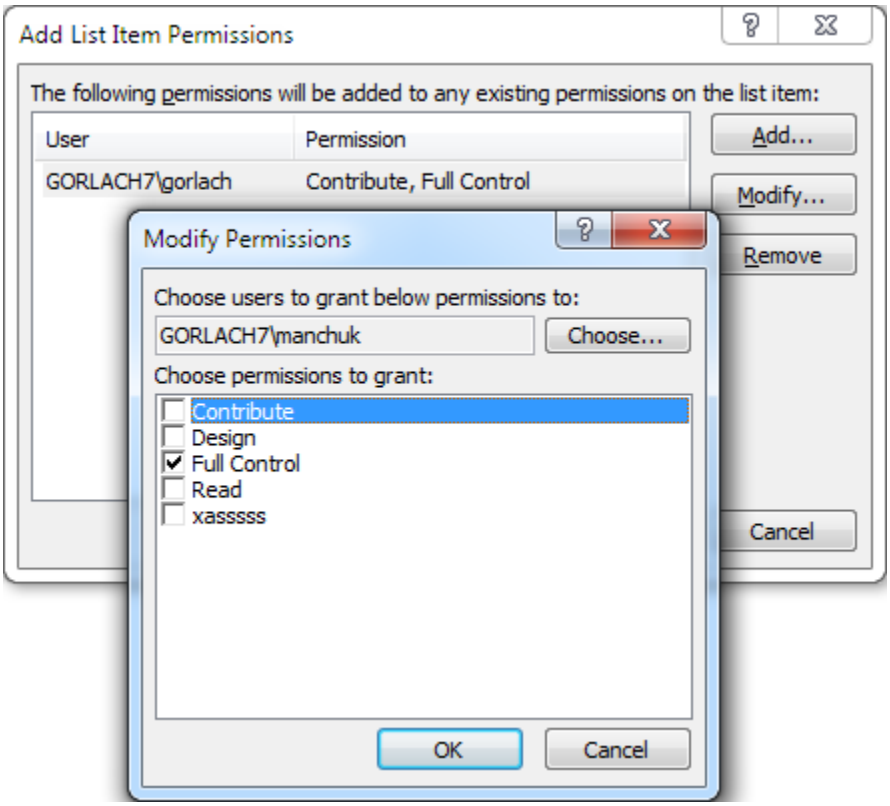
Parameters:

specified - Special-type array of the users and they permissions.
this list - Target list.

13.12 Delete Permissions from List Item

Delete specified permissions from this list item

This workflow actions is used to remove permissions to site users for specified list item. In the Remove List Item Permissions window you can choose multiple users and select any permissions for them including Out Of The Box and custom permissions.



Note: May need to place this action into Impersonation Step. Otherwise, action will be executed under context of the user who started the workflow.

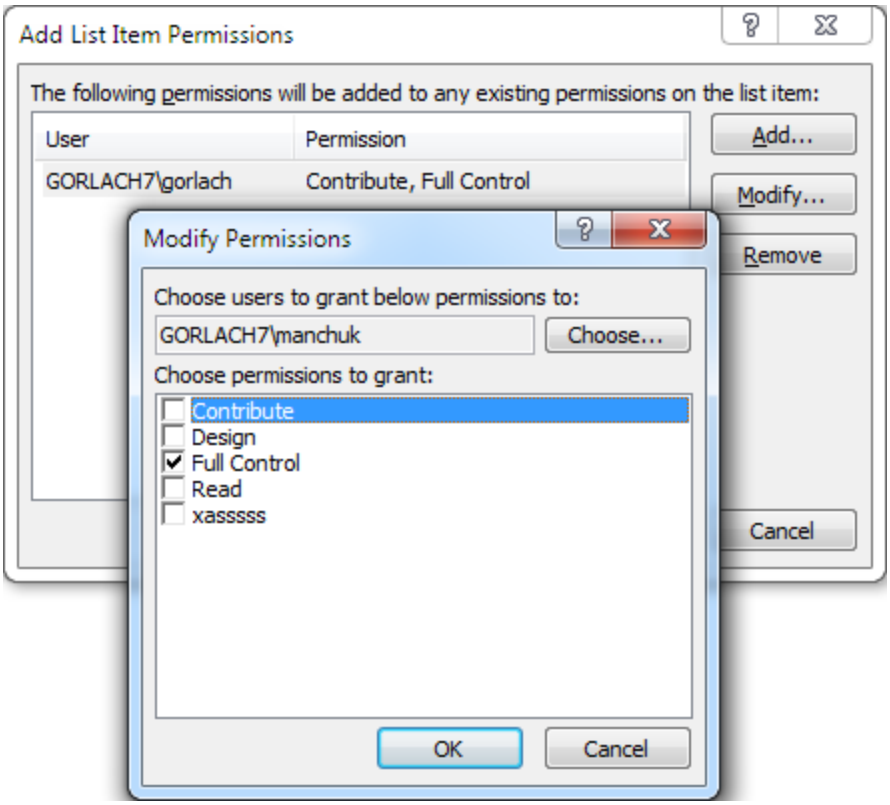
Parameters:

specified - Special-type array of the users and they permissions.
this list item - Target list item.

13.13 Grant Permissions at Url

Grant specified permissions at this url

This workflow actions is used to grant permissions to site users on SharePoint site, list, list folder or list item specified by URL. In the Add List Item Permissions window you can choose multiple users and add any permissions for them including Out Of The Box and custom permissions.



Note: May need to place this action into Impersonation Step. Otherwise, action will be executed under context of the user who started the workflow.

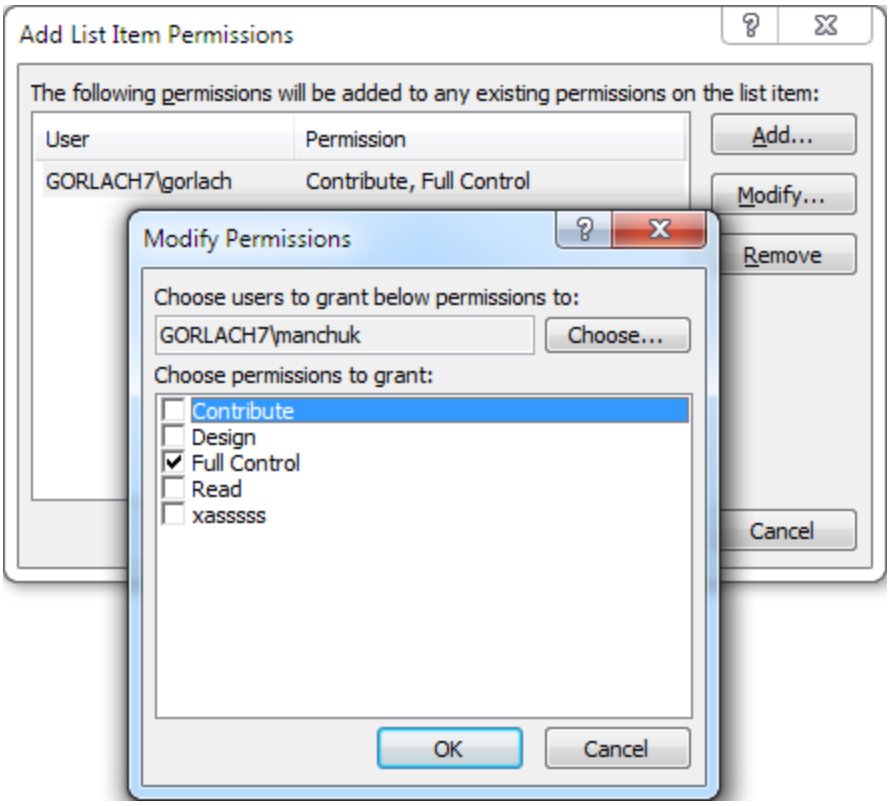
Parameters:

specified - Special-type array of the users and they permissions.
this url - Full URL of a SharePoint site, list, list folder or list item.

13.14 Grant Permissions on List

Grant specified permissions on this list

This workflow actions is used to grant permissions to site users for specified list. In the Add List Item Permissions window you can choose multiple users and add any permissions for them including Out Of The Box and custom permissions.



Note: May need to place this action into Impersonation Step. Otherwise, action will be executed under context of the user who started the workflow.

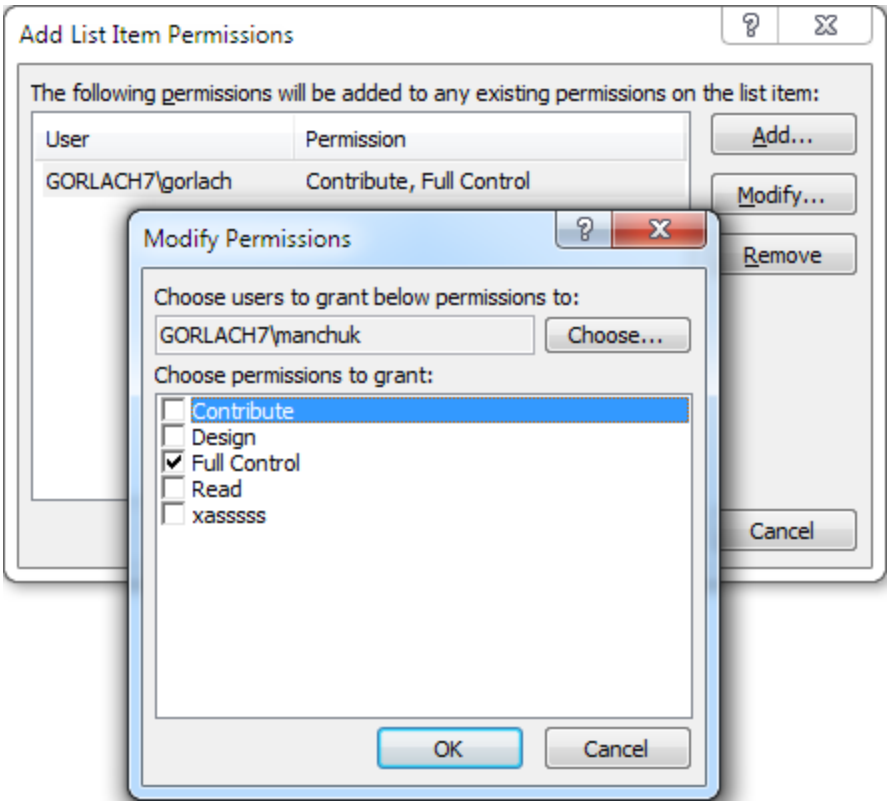
Parameters:

specified - Special-type array of the users and they permissions.
this list - Target list.

13.15 Grant Permissions on List Item

Grant specified permissions on this list item

This workflow actions is used to grant permissions to site users for specified list item. In the Add List Item Permissions window you can choose multiple users and add any permissions for them including Out Of The Box and custom permissions.



Note: May need to place this action into Impersonation Step. Otherwise, action will be executed under context of the user who started the workflow.

Parameters:

specified - Special-type array of the users and they permissions.
this list item - Target list item.

14. Printing activities

14.1 Insert Watermark into PDF Before Printing

Insert a watermark (text or image url) in PDF file that will be printed. Rotation: Diagonal, Opacity: 50, Page range: all. Font name Courier, size 53, color: 00000000, style: Regular

This workflow action adds a watermark to PDF file, that will be printed by the action *'Print document'* that should follow after this action.

Parameters:

text - watermark text.
image url - watermark picture url.
Either of these parameters should be specified. If both parameters specified, **image url** parameter will be used.
Rotation: Diagonal - available values:
Diagonal - rotated 45 degrees clockwise **ReverseDiagonal** - rotated 45 degrees counterclockwise **none** - no rotation
Opacity: 50 - opacity in percents from 0% to 100%.
Page range: all - page range where watermarks should be added.
Possible values:
all - for all pages **1-5, 8, 11-13** - to print pages 1,2,3,4,5,8,11,12,13
Font name: Courier - font family name
Note: this parameter takes effect only for **text** watermarks
size: 53 - font size in points
Note: this parameter takes effect only for **text** watermarks
color: FF000000 - black color in 16-bit ARGB color space
Note: this parameter takes effect only for **text** watermarks
style: Regular - font style.
Available values: **Regular, Bold, Italic, Underline, Strikeout**.
Note: this parameter takes effect only for **text** watermarks

14.2 Print Document

Print document at this url to default printer.

This workflow action is used to print specified document or image to specified printer. List of supported formats: Open XML Document (.docx, .docm, .dotx, .dotm); Word 97-2003 Document (.doc, .dot); Rich Text Format (.rtf); Web Page (.htm, .html, .mht, .mhtml); Text documents (.xml); Images (.jpg, .tif, .png, .gif, .bmp).

Note:To use this action you must install HarePoint Printing Service and configure HarePoint Printing Service Application. See product manual for details.

See Also: "Set Printing Options" action.

Parameters:

this url - Full URL of the document.
default - Network name of the printer.

Examples:

Print document
at http://sp14.vlab.mapilab.local:133/Shared%20Documents/Test.docx to \\KOENIG\OKIB410 printer.

14.3 Set Printing Options

Set number of copies: 1, print range: all and orientation: portrait.

This workflow action is used to specify page orientation, number of copies and pages to print.

Parameters:

1 - Specifies number of copies to print.
all - Specfies pages range to print, e.g. 1-5, 8, 11-13.
portrait - Specifies page layout.

15. Social activities

15.1 Create Group in Yammer

Create new public group in Yammer with this title, this description, use this access code

This workflow action creates a new group in Yammer with specified title and description. The group can be public or private

Parameters:

public group - choose if the new group should be *public* or *private*.
this title - specify the title for the group
this description - specify description for the group
use **this access code** - specify the access code:
There are two ways to receive it:

- Using **Social Authorize Tool** from **Client Components** of HarePoint Workflow Extensions (use Installer to install Client Components)
- Using the following link: http://auth.mapilab.com/Yammer/RequestToken

15.2 Create MailChimp Campaign

Create a new MailChimp campaign from html with this subject and this body to recipient name and recipient emails list. Use this api key, store campaign id in Variable: variable.

This workflow action is used to create a new MailChimp campaign.

Note: this action only creates a campaign, but doesn't send it.

See also: "Send MailChimp campaign" action.

Parameters:

html/plain text - select if e-mail body should be in HTML or Plain Text.
this subject - the e-mail subject, and also campaign name on MailChimp site.
this body - the e-mail body
recepient name - (Optional) the recepient name for "*To:*" field.
Macros are supported:

- E-mail address *|EMAIL|*
- First name *|FNAME|*

- Last name *|LNAME|*
- Unsubscribe link *|UNSUB|*
- Forward to a Friend Link *|FORWARD|*
- Update Profile Link *|UPDATE_PROFILE|*
- MonkeyRewards Affiliate Link *|REWARDS|*

recipient emails list - the name of the recepient e-mails list from MailChimp site. This list contains e-mails of the recipients for this campaign.

this api key - the authorization key. See http://kb.mailchimp.com/accounts/management/about-api-keys

store campaign id in Variable:variable - specify the variable where an ID for this campaign will be stored.

15.3 Create User in Yammer

Create user in Yammer with this e-mail, this full name, this job title, this department, use this access code

This workflow action creates a new user in Yammer network with specified parameters

Parameters:

this e-mail - specify the e-mail address for the new user

this full name - specify the full name for the new user

this job title - specify the job title for the new user

this department - specify the department for the new user

use **this access code** - specify the access code

There are two ways to receive it:

- Using **Social Authorize Tool** from **Client Components** of HarePoint Workflow Extensions (use Installer to install Client Components)
- Using the following link: http://auth.mapilab.com/Yammer/RequestToken

15.4 Delete MailChimp Campaign

Delete this MailChimp campaign permanently. Use this api key.

This workflow action is used to delete MailChimp campaign permanently.

Important note: the deleted campaign cannot be restored!

Parameters:

this MailChimp campaign - specify either *ID* or *name* of the campaign to be deleted.

Note: If the name is specified and there are multiple campaigns with this name, the latest created campaign will be removed.

this api key - the authorization key. See http://kb.mailchimp.com/accounts/management/about-api-keys

15.5 Delete User from Yammer

Delete user with this e-mail from Yammer, use this access code

This workflow action deletes the specified user from Yammer

Parameters:

user with this e-mail - specify the e-mail address of the user

use **this access code** - specify the access code

There are two ways to receive it:

- Using **Social Authorize Tool** from **Client Components** of HarePoint Workflow Extensions (use Installer to install Client Components)
- Using the following link: http://auth.mapilab.com/Yammer/RequestToken

15.6 Get User's Property from Yammer

Get specified property of Yammer user with this e-mail address, use this access code, store result in Variable: variable

This workflow action gets the specified property from the profile of the specified user in Yammer

Parameters:

specified property - specify the profile property to get
user with this e-mail address - specify the user's e-mail address
use **this access code** - specify the access code:
There are two ways to receive it:
• Using **Social Authorize Tool** from **Client Components** of HarePoint Workflow Extensions (use Installer to install Client Components)
• Using the following link: http://auth.mapilab.com/Yammer/RequestToken
store result in Variable: variable - specify the variable to store the output

15.7 Invite User to Yammer

Send Yammer invitation e-mail to this e-mail address, use this access code

This workflow action initiates sending an invitation e-mail from Yammer network to specified e-mail address

Parameters:

this e-mail address - specify the recipient's e-mail address
use **this access code** - specify the access code:
There are two ways to receive it:
• Using **Social Authorize Tool** from **Client Components** of HarePoint Workflow Extensions (use Installer to install Client Components)
• Using the following link: http://auth.mapilab.com/Yammer/RequestToken

15.8 Publish Message to Social Media via dlvr.it

Publish message with this text, this title and attach image at this url using this dlvr.it API key and this route

This workflow action allows posting to Facebook, Twitter, Google+ and other social networks using dlvr.it.

Note: This action requires registration on dlvr.it.

Free subscription allows posting to 3 networks.

Parameters:

this text - message text
this title - message title (optional)
attach image at this url - path to an image to be attached to a post (optional)
this API key - user's secret key to access dlvr.it API
API key can be checked here: https://app.dlvr.it/settings/account
this route - route name for dlvr.it that will be used to post message
The route is set up in dlvr.it (https://app.dlvr.it/deliveries) right after you log into it. Add there RSS, then add those social networks that you wish to repost to.
Specify the route name in this parameter.

15.9 Publish Message to Yammer

Publish this message to All Company Yammer group, attach file at this url, use this access code.

This workflow action posts a message to any group of Yammer corporate network; a file can be attached from the specified document library.

Parameters:

this message - a text of the message to be posted
to **All Company** Yammer group - select the Yammer group where this message should be posted
attach **file at this url** - specify the URL of the file from Document Library in SharePoint
use **this access code** - specify the access code:
There are two ways to receive it:

- Using **Social Authorize Tool** from **Client Components** of HarePoint Workflow Extensions (use Installer to install Client Components)
- Using the following link: http://auth.mapilab.com/Yammer/RequestToken

15.10 Replicate MailChimp Campaign

Replicate this MailChimp campaign. Use this api key. Store replicated campaign id in Variable: variable.

This workflow action is used to replicate (create a copy of) an existing MailChimp campaign.

Parameters:

this MailChimp campaign - specify either *ID* or *name* of the campaign to be replicated.

Note: If the name is specified and there are multiple campaigns with this name, the latest created campaign will be replicated.

this api key - the authorization key. See http://kb.mailchimp.com/accounts/management/about-api-keys

store campaign id in Variable:variable - specify the variable where an ID for the new campaign will be stored.

15.11 Schedule MailChimp Campaign

Schedule this MailChimp campaign for this time. Use this api key.

This workflow action is used to schedule the MailChimp campaign to be sent on a certain date at a certain time.

Note: A campaign can also be sent immediately, see "Send MailChimp Campaign" action.

Parameters:

this MailChimp campaign - specify either *ID* or *name* of the campaign to be scheduled.

Note: If the name is specified and there are multiple campaigns with this name, the latest created campaign will be scheduled.

this time - specify the time in GMT (Greenwhich Mean Time) time zone, in 24h format as follows:

yyyy-mm-dd hh:mm:ss

Example:

2013-12-30 20:30:00

this api key - the authorization key. See http://kb.mailchimp.com/accounts/management/about-api-keys

15.12 Send MailChimp Campaign

Send this MailChimp campaign immediately. Use this api key.

This workflow action is used to initate sending a MailChimp campaign immediately.

Note: The campaign sending can also be scheduled - see "Schedule MailChimp Campaign" action.

Parameters:

this MailChimp campaign - specify either *ID* or *name* of the campaign to be sent.

Note: If the name is specified and there are multiple campaigns with this name, the latest created campaign will be sent.

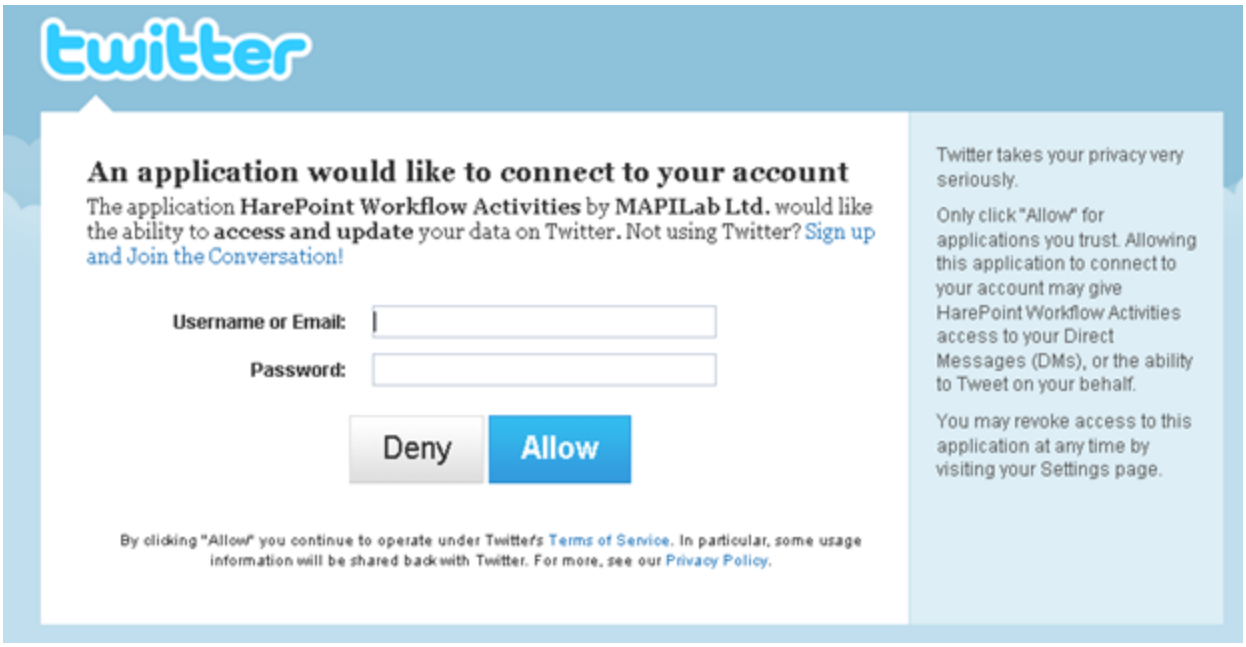
this api key - the authorization key. See http://kb.mailchimp.com/accounts/management/about-api-keys

15.13 Send Twitter Message

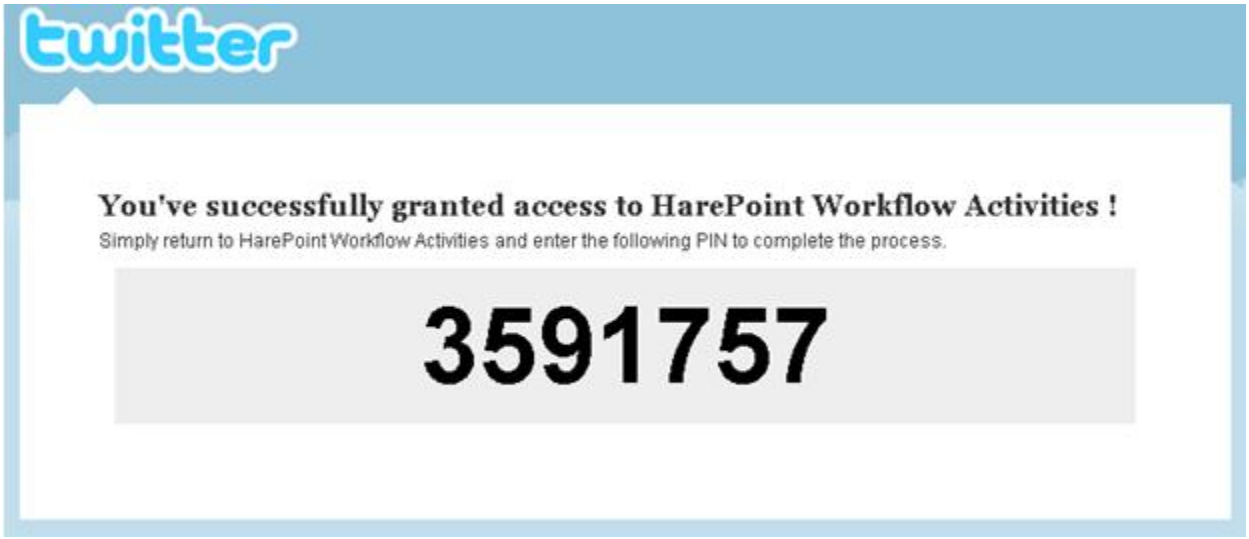
Send message with specified twitter authorization token.

This workflow action is used to send message to twitter. To set up this action, you must create authorization token with "Twitter Authorization Tool" placed in the program folder in Start menu.

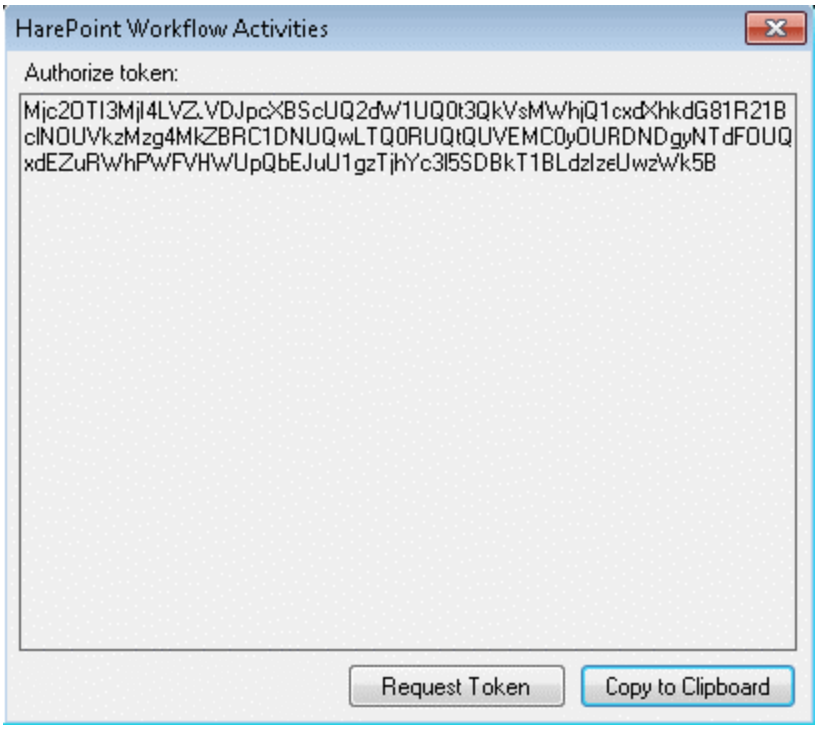
Run the tool, and click "Request Token" button. After that, Twitter web-page will be opened:


twitter
An application would like to connect to your account
The application HarePoint Workflow Activities by MAPILab Ltd. would like the ability to access and update your data on Twitter. Not using Twitter? Sign up and Join the Conversation!
Username or Email:
Password:
Deny
Allow
By clicking "Allow" you continue to operate under Twitter's Terms of Service. In particular, some usage information will be shared back with Twitter. For more, see our Privacy Policy.
Twitter takes your privacy very seriously.
Only click "Allow" for applications you trust. Allowing this application to connect to your account may give HarePoint Workflow Activities access to your Direct Messages (DMs), or the ability to Tweet on your behalf.
You may revoke access to this application at any time by visiting your Settings page.

Enter your credentials and press Allow button on the web page to get the PIN number:


twitter
You've successfully granted access to HarePoint Workflow Activities !
Simply return to HarePoint Workflow Activities and enter the following PIN to complete the process.
3591757

Enter the PIN number to "Twitter Authorization Tool" and get the authorization token:



Press "Copy to Clipboard" button and paste authorization token to the second parameter of this action.

Parameters:

message - Message text. Maximum length is 140 characters, longer messages will be cut off automatically.
specified - Twitter authorization token generated with "Twitter Authorization Tool".

15.14 Set User's Property in Yammer

Set specified property of Yammer user with this e-mail address to this value, use this access code

This workflow action sets the specified property to the profile of the specified user in Yammer

Parameters:

specified property - select the property to set
user with this e-mail address - specify the user's e-mail address
this value - specify the new value for this property
use **this access code** - specify the access code:

There are two ways to receive it:

- Using **Social Authorize Tool** from **Client Components** of HarePoint Workflow Extensions (use Installer to install Client Components)
- Using the following link: http://auth.mapilab.com/Yammer/RequestToken

15.15 Subscribe to MailChimp List

Subscribe this e-mail address to this MailChimp list. Use this api key.

This workflow action is used to add a new subscriber to a MailChimp list.

Parameters:

this e-mail address - specify the e-mail address of a new subscriber.

this MailChimp list - specify the name of the MailChimp list where the new subscriber needs to be added.

this api key - the authorization key. See http://kb.mailchimp.com/accounts/management/about-api-keys

15.16 Unsubscribe from MailChimp List

Unsubscribe this e-mail address from this MailChimp list. Use this api key, delete completely: no, send goodbye: yes, notify: yes .

This workflow action is used to unsubscribe people from your MailChimp list.

Parameters:

this e-mail address - specify e-mail address of a person that needs to be unsubscribed.

this MailChimp list - specify the name of the MailChimp list from which the subscriber should be removed.

this api key - the authorization key. See http://kb.mailchimp.com/accounts/management/about-api-keys

delete completely: no/yes - select if the subscriber should be removed completely from the list, or only marked as unsubscribed.

send goodbye: yes/no - specify if an e-mail should be sent to a person, informing that he or she is now unsubscribed.

notify: yes/no - specify if an e-mail should be sent to the administrator of the MailChimp list informing that a user has been unsubscribed.

15.17 Update Group in Yammer

Update this group in Yammer, set this title, this description, make it public, use this access code

This workflow action modifies the parameters of the existing group in Yammer

Parameters:

this group - specify the group to be updated

this title - specify the new title for the group

this description - specify the new description for the group

make it public - choose to make this group *public* or *private*

use **this access code** - specify the access code:

There are two ways to receive it:

- Using **Social Authorize Tool** from **Client Components** of HarePoint Workflow Extensions (use Installer to install Client Components)
- Using the following link: http://auth.mapilab.com/Yammer/RequestToken

16. Strings activities

16.1 Capture All Occurrences of Text

Capture regular expression in this text and store all occurrences in Variable: MatchResults

This workflow action is used to store all matches of regular expression in specified text to output array variable.

Regular expressions provide a powerful, flexible, and efficient method for processing text. To learn more about regular expressions, see: http://msdn.microsoft.com/en-us/library/hs600312.aspx

For regular expression language reference, see: http://msdn.microsoft.com/en-us/library/az24scfc.aspx

Note: To set additional options for this activity, like case sensitivity, use 'Set Regular Expression Options' activity.

Parameters:

regular expression - Regular expression pattern. For example, like '\b(\w+)\s\1\b'.
this text - Input text, expression, item field or variable.
Variable: MatchResults - Output array variable, each item contains one match of regular expression.

Examples:

Capture \b[A-Z0-9._%-]+@[A-Z0-9.-]+\.[A-Z]{2,4}\b in xxx kkk@my.com owkk@ @kkk s@i12.com ddd and store all occurrences in Variable: MatchResults2

then Join Variable: MatchResults2 at index 0 next -1 items using ; and store in Variable: emails

then Log Variable: emails to the workflow history list

Result in workflow history list:

kkk@my.com; s@i12.com

16.2 Capture Text

Capture regular expression in this text and store in Variable: FirstMatchResult

This workflow action is used to store first match of regular expression in specified text to output text variable.

Regular expressions provide a powerful, flexible, and efficient method for processing text. To learn more about regular expressions, see: http://msdn.microsoft.com/en-us/library/hs600312.aspx

For regular expression language reference, see: http://msdn.microsoft.com/en-us/library/az24scfc.aspx

Note: To set additional options for this activity, like case sensitivity, use 'Set Regular Expression Options' activity.

Parameters:

regular expression - Regular expression pattern. For example, like '\b(\w+)\s\1\b'.
this text - Input text, expression, item field or variable.
Variable: FirstMatchResult - Output string variable with the text of the first match of regular expression.

Examples:

Set regular expression options: ignore case - No, is pattern culture invariant - Yes, use multiline mode - No, use singleline mode - No, search right to left - No

then Capture .est in IT IS TEST and store in Variable: r1

then Set regular expression options: ignore case - Yes, is pattern culture invariant - Yes, use multiline mode - No, use singleline mode - No, search right to left - No

then Capture .est in IT IS TEST and store in Variable: r2

then Log <[%Variable: r1%]>, <[%Variable: r2%]> to the workflow history list

Result in workflow history list:

<>, <TEST>

16.3 Convert HTML To Text

Convert this HTML code to plain text and store result in Variable: variable.

This workflow action is used to strip HTML tags from specified string and return result in to output variable.

Parameters:

this - Text, expression, item field or variable to convert.
Variable: variable - Output string variable.

16.4 Convert Number To Text

Convert this number to text in English and store result in Variable: variable.

This workflow action is used to convert integer number into text using specified language. Supported languages: English, French, German, Russian.

Parameters:

this number - Input integer to convert.
English - One of the available languages.

Variable: variable - Output string variable.

Examples:

Convert Variable: totalPrice number to text in English and store result in Variable: totalPriceText.

16.5 Convert Text Case

Convert this text to uppercase and store in Variable: variable.

Converts specified text to UPPER CASE, lower case, Title Case or Sentence case.

Parameters:

this text - Text, expression, item field or variable to convert.
uppercase - One of the following conversion types: *uppercase, lowercase, title case, sentence case*.
Variable: variable - Output string variable.

16.6 Converts Base64 to Text

Converts specified Base64 text to text using default encoding and store result in Variable: ConvertedText.

This workflow activity will decode Base64 string, interpret resulting data as the string in one from five available encodings, and copy to output system string variable.

Available encodings:

- default - 8-bit encoding of the current operating system's ANSI code page.
- ASCII - 7-bit ASCII encoding.
- Unicode - also known as UTF16, represents each character as a sequence of one or two 16-bit integers.
- UTF8 - uses one byte for any ASCII characters, which have the same code values in both UTF-8 and ASCII encoding, and up to four bytes for other characters.
- UTF32 - uses exactly 32 bits for each character.

Because all Default encodings and ANSI encoding lose data (see Examples section), your workflow might use UTF8 instead.

Parameters:

specified Base64 text - Input text, expression, item field or variable.
default encoding - One from five available encodings.
Variable: ConvertedText - Output string variable.

Examples:

Let's encode German word 'Grüß' with different encodings with 'Converts Text to Base64' activity on the machine with Russian ANSI code page and decode results back with the same encodings with 'Converts Base64 to Text' activity.

- *default*: encoded string 'R3J1Pw==', decoded string 'Gru '
- *ASCII*: encoded string 'R3I/Pw==', decoded string 'Gr '
- *Unicode*: encoded string 'RwByAPwA3wA=', decoded string 'Grüß'
- *UTF8*: encoded string 'R3LDvMOf', decoded string 'Grüß'
- *UTF32*: encoded string 'RwAAAHIAAAD8AAAA3wAAAA==', decoded string 'Grüß'

As you can see, all three Unicode encodings save the German national characters unchanged. With 7-bit ASCII encoding, all national German characters were lost. In the case of default encoding, input string was initially mapped to Russian ANSI code page, where the German 'ü' was mapped to latin 'u' and the last 'ß' was lost, because it has no suitable mapping to Russian code page.

16.7 Converts Text to Base64

Converts specified text to Base64 using default encoding and store result in Variable: ConvertedText1.

This workflow activity will encode specified text with one from five available encodings and convert encoded text to Base64.

Available encodings:

- default - 8-bit encoding of the current operating system's ANSI code page.
- ASCII - 7-bit ASCII encoding.
- Unicode - also known as UTF16, represents each character as a sequence of one or two 16-bit integers.

- UTF8 - uses one byte for any ASCII characters, which have the same code values in both UTF-8 and ASCII encoding, and up to four bytes for other characters.
- UTF32 - uses exactly 32 bits for each character.

Because all Default encodings and ANSI encoding lose data (see Examples section), your workflow might use UTF8 instead.

Parameters:

specified text - Text, expression, item field or variable to encode.
default encoding - One from five available encodings.
Variable: ConvertedText1 - Output string variable.

Examples:

Let's encode German word 'Grüß' with different encodings with 'Converts Text to Base64' activity on the machine with Russian ANSI code page and decode results back with the same encodings with 'Converts Base64 to Text' activity.

- *default*: encoded string 'R3J1Pw==', decoded string 'Gru '
- *ASCII*: encoded string 'R3I/Pw==', decoded string 'Gr '
- *Unicode*: encoded string 'RwByAPwA3wA=', decoded string 'Grüß'
- *UTF8*: encoded string 'R3LDvMOf', decoded string 'Grüß'
- *UTF32*: encoded string 'RwAAAHIAAAD8AAAA3wAAAA==', decoded string 'Grüß'

As you can see, all three Unicode encodings save the German national characters unchanged. With 7-bit ASCII encoding, all national German characters were lost. In the case of default encoding, input string was initially mapped to Russian ANSI code page, where the German 'ü' was mapped to latin 'u' and the last 'ß' was lost, because it has no suitable mapping to Russian code page.

16.8 Decode Web Text

Decode specified text using url decoder and store in Variable: DecodedText.

This workflow action will decode text with one from two available decoding methods and return result in to output variable.

Available decode methods:

1. *url decoder* - Converts a URL-encoded string into a decoded string. If characters such as blanks and punctuation are passed in an HTTP stream, they might be misinterpreted at the receiving end. URL encoding converts characters that are not allowed in a URL into character-entity equivalents; URL decoding reverses the encoding. For example, when embedded in a block of text to be transmitted in a URL, the characters < and > are encoded as %3c and %3e, this method decodes its back.

2. *html decoder* - Converts a string that has been HTML-encoded for HTTP transmission into a decoded string. If characters such as blanks and punctuation are passed in an HTTP stream, they might be misinterpreted at the receiving end. HTML encoding converts characters that are not allowed in HTML into character-entity equivalents; HTML decoding reverses the encoding. For example, when embedded in a block of text, the characters < and > are encoded as < and > for HTTP transmission, this method decodes its back.

Parameters:

specified text - Text, expression, item field or variable to decode.
url decoder - One from two decoding methods: *url decoder* or *html decoder*.
Variable: DecodedText - Output string variable.

16.9 Encode Web Text

Encode specified text using url encoder and store in Variable: EncodedText.

Process text with one from four available encoding methods and returns result to output variable.

Available encoding methods:

1. *url encoder* - Can be used to encode the entire URL, including query-string values. If characters such as blanks and punctuation are passed in an HTTP stream, they might be misinterpreted at the receiving end. URL encoding converts characters that are not allowed in a URL into character-entity equivalents. For example, when the characters < and > are embedded in a block of text to be transmitted in a URL, they are encoded as %3c and %3e. Note, that this method converts each space character to a plus character (+).

2. *url path encoder* - The same as above, but converts each space character into the string "%20", which represents a space in hexadecimal notation. Use the this method when you encode the path portion of a URL in order to guarantee a consistent decoded URL, regardless of which platform or browser performs the decoding.

3. *html encoder* - Converts a string to an HTML-encoded string. If characters such as blanks and punctuation are passed in an HTTP stream, they might be misinterpreted at the receiving end. HTML encoding converts characters that are not allowed in HTML into character-entity equivalents. For example, when embedded in a block of text, the characters < and > are encoded as < and > for HTTP transmission.

4. *html attribute encoder* - Minimally converts a string to an HTML-encoded string. It converts only quotation marks ("), ampersands (&), and left angle brackets (<) to equivalent character entities. It is considerably faster than the *html encoder*. The result string should be used only for double-quoted (") attributes, security issues might arise when using result string with single-quoted (') attributes.

Parameters:

specified text - Text, expression, item field or variable to encode.
url encoder - One from four available encoding methods: *url encoder, url path encoder, html encoder, html attribute encoder.*
Variable: EncodedText - Output string variable.

Examples:

Encode [%Current Item:Approver Comments%] using url encoder and store inVariable: param.

then Send HTTP request to http://mysite/ q=[%Variable: param%] using GET method with this headers. Store response in Variable: pageBody and status code in Variable: code.

16.10 Evaluate Mathematical Expression

Evaluate this mathematical expression and store result in variable

This workflow action is used to evaluate the specified mathematical expression. In particular, this action can be used to add up values from multiple list columns.

Parameters:

this - specify the mathematical expression.
Supported operators: + - * /
Note: there are more supported operators and functions, for more details, please refer to https://ncalc.codeplex.com/
variable - specify the variable to keep the result.

Examples:

Evaluate [%Current Item: File Size%] + 100 mathematical expression and store result in Variable: test integer

16.11 Extract Substring

Extract substring from this text, starting at index 0 with -1 length and store in Variable: Substring.

Retrieves a substring from input text. The substring starts at a specified character position and has a specified length.

The main difference from standard SharePoint activity "Extract Substring from String from Index with Length" is following. In the case of exit out of bounds of string, standard activity will return empty string when this activity will decrease length parameter according with length of string.

Parameters:

this text - Input text, expression, item field or variable.
index 0 - Zero-based character position where substring starts.
-1 length - Substring length. If the length is out of string bounds, it will be decreased according with length of string. -1 is a special value which means 'to the end of input string'.
Variable: Substring - Output string variable.

Examples:

1. Example with this activity:

Extract substring from 123456, starting at index 3 with 2 length and store in Variable: var1.

then Extract substring from 123456, starting at index 1 with -1 length and store in Variable: var2.

then Extract substring from 123456, starting at index 1 with 100 length and store in Variable: var3.

Result: var1 is '45', var2 is '23456', var3 is '23456'.

2. Example with standard SharePoint activity "Extract Substring from String from Index with Length":

Copy from 123456, starting at 3 for 2 characters (Output to Variable: var1)

then Copy from 123456, starting at 1 for 100 characters (Output to Variable: var2)

Result: var1 is '45', var2 is ''.

16.12 Format Field of Current Item

Format this field with this format and store in Variable: FormattedText.

This workflow action is used to convert current item field to string with special formatting.

Format sting may have dozens of different specifiers. You can learn more about formatting is the following articles:

- Formatting Types: http://msdn.microsoft.com/en-us/library/26etazsy.aspx#NetFormatting
- Composite Formatting: http://msdn.microsoft.com/en-us/library/txafckwd.aspx
- Standard Date and Time Format Strings: http://msdn.microsoft.com/en-us/library/az4se3k1.aspx
- Standard Numeric Format Strings: http://msdn.microsoft.com/en-us/library/dwhawy9k.aspx
- Custom Numeric Format Strings: http://msdn.microsoft.com/en-us/library/0c899ak8.aspx

Parameters:

this field - Input string with the name of Current Item field.
this format - Format string.
Variable: FormattedText - Output string variable.

Examples:

Format Created field with {0:yyyy-MM-dd} and store in Variable: F2

then Format Modified By field with Item was modified by: {0} and store in Variable: F3

then Log [%Variable: F3%] at [%Variable: F2%] to the workflow history list

The result in workflow history list:

Item was modified by: 1;#GORLACH7\gorlach at 2011-04-30

16.13 Get Random Number

Get random number within the range from 0 to 100 and store in Variable: RandomNumber1.

This workflow action is used to get the random number from specified range.

Note: The range of return values includes lower limit value but not higher limit value.

See Also: "Get Random String" activity.

Parameters:

0 - The lower limit of the range. May be negative.

100 - The higher limit of the range. May be negative.

Variable: RandomNumber1 - Output integer variable.

16.14 Get Random String

Get random string 16 characters long containing alphanumeric symbols and store in Variable:RandomString.

This workflow action is used to get the random string which contains characters from specified set.

By default, the following set is used: abcdefghijklmnopqrstuvwxyz0123456789

See Also: "Get Random Number" activity.

Parameters:

16 - Output string length.

alphanumeric - The set of characters for output string.

Variable:RandomString - Output string variable.

16.15 Is Value Matches Regular Expression (HarePoint)

If value matches regular expression (ignore case: Yes, is pattern culture invariant: Yes, use multiline mode: No, use singleline mode: No, search right to left: No)

This workflow condition returns True is the specified value matches with regular expression.

Regular expressions provide a powerful, flexible, and efficient method for processing text. To learn more about regular expressions, see: http://msdn.microsoft.com/en-us/library/hs600312.aspx

For regular expression language reference, see: http://msdn.microsoft.com/en-us/library/az24scfc.aspx

Parameters:

value - Text, expression or variable to check.
regular expression - Regular expression pattern. For example, like '\b(\w+)\s\1\b'.
ignore case - Ignore case in regular expression and in the input string.
pattern is culture invariant- Specifies that cultural differences in language is ignored. Culture is used to obtain culture-specific information, such as casing rules and the alphabetical order of individual characters.
multiline mode- Multiline mode. Changes the meaning of ^ and $ so they match at the beginning and end, respectively, of any line, and not just the beginning and end of the entire string.
singleline mode- Specifies single-line mode. Changes the meaning of the dot (.) so it matches every character (instead of every character except \n).
search right to left - Specifies that the search will be from right to left instead of from left to right.

16.16 Join Text

Joins string array at index 0 next -1 items using specified delimiter and store in Variable: JoinedText

This workflow action is used to join strings from array to one string using specified delimiter.

See Also: "Split Text" activity.

Parameters:

string array - Input strings array.
index 0 - Zero-based index of the first array item to join.
next -1 - Zero-based index of the last array item to join. -1 means 'to the end of the array'.
specified delimiter - String placed between two array items.
Variable: JoinedText - Output string variable.

Examples:

Capture \b[A-Z0-9._%-]+@[A-Z0-9.-]+\.[A-Z]{2,4}\b in xxx kkk@my.com owkk@ @kkk s@i12.com ddd and store all occurrences inVariable: MatchResults2

then Join Variable: MatchResults2 at index 0 next -1 items using ; and store in Variable: emails

then Log Variable: emails to the workflow history list

Result in workflow history list:

kkk@my.com; s@i12.com

16.17 Replace Text

Find regular expression in this text, replace with this text and store in Variable: ReplaceResult.

This workflow action is used to store replace all matches of regular expression in specified text with specified string.

Regular expressions provide a powerful, flexible, and efficient method for processing text. To learn more about regular expressions, see: http://msdn.microsoft.com/en-us/library/hs600312.aspx

For regular expression language reference, see: http://msdn.microsoft.com/en-us/library/az24scfc.aspx

Note: To set additional options for this activity, like case sensitivity, use 'Set Regular Expression Options' activity.

Parameters:

regular expression - Regular expression pattern. For example, like '\b(\w+)\s\1\b'.
this text - Input text, expression, item field or variable.
replace with this text - String to replace all matches.
Variable: ReplaceResult - Output string variable.

16.18 Set Regular Expression Options

Set regular expression options: ignore case - Yes, is pattern culture invariant - Yes, use multiline mode - No, use singleline mode - No, search right to left -No.

This workflow actions set options for regular expression activities like 'Capture Text' and 'Replace Text' in the current workflow step.

Parameters:

ignore case - Ignore case in regular expression and in the input string.
pattern is culture invariant - Specifies that cultural differences in language is ignored. Culture is used to obtain culture-specific information, such as casing rules and the alphabetical order of individual characters.
multiline mode - Multiline mode. Changes the meaning of ^ and $ so they match at the beginning and end, respectively, of any line, and not just the beginning and end of the entire string.
singleline mode - Specifies single-line mode. Changes the meaning of the dot (.) so it matches every character (instead of every character except \n).
search right to left - Specifies that the search will be from right to left instead of from left to right.

Examples:

Set regular expression options: ignore case - No, is pattern culture invariant - Yes, use multiline mode - No, use singleline mode - No, search right to left - No

then Capture .est in IT IS TEST and store in Variable: r1

then Set regular expression options: ignore case - Yes, is pattern culture invariant - Yes, use multiline mode - No, use singleline mode - No, search right to left - No

then Capture .est in IT IS TEST and store in Variable: r2

then Log <[%Variable: r1%]>, <[%Variable: r2%]> to the workflow history list

Result in workflow history list:

<>, <TEST>

16.19 Split Text

Splits specified text using specified delimiters (strings, one per line) and store in Variable: StringsArray

This workflow action is used to split text and store results to strings array. You can specify string delimiters (one per line) or character delimiters (each charachter in specified string is delimiter).

See Also: "Join Text" activity.

Parameters:

specified text - Text, expression, item field or variable.

specified delimiters - String or multi-line string with delimiters.

strings/characters - Type of delimiters. You can choose string delimiters (one per line) or character delimiters (each charachter in specified string is delimiter).

Variable: StringsArray - Output strings array.

16.20 Trim Text

Removes leading and trailing white-space characters from this text and store in Variable: TrimmedText

Removes white-space characters from beginning and/or end of specified text.

Parameters:

leading and trailing - One from three options: *leading*, *trailing*, *leading and trailing*.

this text - Text, expression, item field or variable.

Variable: TrimmedText - Output variable.

17. Workflow activities

17.1 Get Workflow Association Data

Get this workflow default association data and store in Variable: variable

This workflow action is used to retrieve workflow association data.

See Also: "Start Workflow" activity.

Parameters:

this workflow - Workflow name.

Variable: variable - Output string variable with workflow association data in XML format.

17.2 Get Workflow Status

Get status of workflow with this name started on item at this url, store status in Variable: variable1, last status change time in Variable: variable2.

This workflow action is used to get site or list workflow status and the time of last status chnage.

Known workflow status strings are:

- Not Started
- Failed On Start
- In Progress
- Error Occurred
- Stopped By User
- Completed
- Canceled

This list may be extended by custom values, e.g. using 'Set Workflow Status' action.

Parameters:

with this workflow name - Workflow name, e.g. 'My workflow'.
item at this url - Optional parameter with item URL.
Variable: variable1 - Output string variable for status code, e.g. 'Completed'.
Variable: variable2 - Output Date/Time variable for last status change time.

17.3 Start Workflow

Start this workflow on the item with this url with default association data and store instance ID in Variable: variable.

This workflow action is used to start list or site workflow on SharePoint site. Due to the nature of Microsoft SharePoint Workflow Timer starting a new workflow can take several seconds to several minutes, during which the workflow will have the status "Starting".

See Also: "Get Workflow Association Data" activity.

Note: This action can be used to start a workflow on multiple list items. See *Example 2* for more details.

Parameters:

this workflow - Workflow name.
the item with this url - Optional parameter with item URL. Arrays of list items URLs are supported, in this case the workflow will start on all list items from the array.
default - Workflow association data in XML format.
Variable: variable - Output string variable for workflow instance identifier.
In case if the action starts multiple workflows, this variable will contain multiple Instance IDs delimited by a semicolon (;).
Example: 1044a581-f7c0-404f-89d5-ae849e39910f;11842b5d-a9a6-4a4b-ac79-d070aaacb96b

Examples:

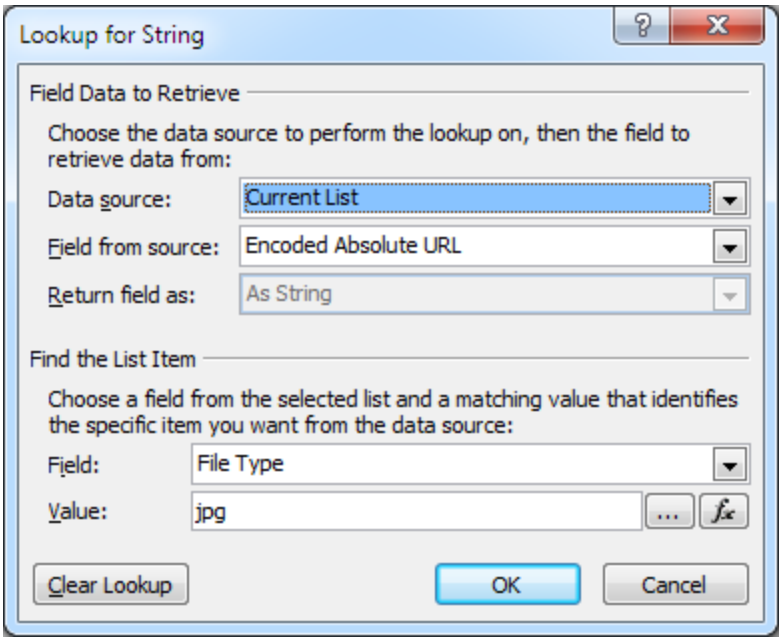
Example 1: Converting all JPG images in document library to PNG

Convert image at Current Item:Encoded Absolute URL to PNG format and store url to converted image in Variable: ConvertedImageUrl. Overwrite exisiting image.

then Delete item at Current Item: Encoded Absolute URL list item url.

then Start ConvertToPng on Current List: Encoded Absolute URL with default association data and store instance ID in Variable: ID.

This sample workflow convert all JPG images in document library to PNG. Use the following lookup settings for the second parameter of 'Start Workflow' action:


Lookup for String
Field Data to Retrieve
Choose the data source to perform the lookup on, then the field to retrieve data from:
Data source:
Current List
Field from source:
Encoded Absolute URL
Return field as:
As String
Find the List Item
Choose a field from the selected list and a matching value that identifies the specific item you want from the data source:
Field:
File Type
Value:
jpg
Clear Lookup
OK
Cancel

Example 2: Starting a workflow on multiple list items

Find item in the list at Current Item:Encoded Absolute URL where item's Title begins with abc . Store top 100 result urls in Variable: array .

then Start compute discount on Variable: array with default association data and store instance ID in Variable: variable.

17.4 Terminate Workflow

Terminate workflow with this instance ID.

This workflow action is used to terminate running workflows started by 'Start Workflow' action.

Parameters:

with this instance ID - The ID of running workflow obtained from 'Start Workflow' action.

17.5 Wait Workflow To Complete

Wait until workflow with this instance ID is completed and store status of the workflow in Variable: variable.

This workflow action is used to wait until workflow started by 'Start Workflow' action is completed.

Known workflow status strings are:

- Not Started
- Failed On Start
- In Progress
- Error Occurred
- Stopped By User
- Completed
- Canceled

This list may be extended by custom values, e.g. using 'Set Workflow Status' action.

Parameters:

with this instance ID - The ID of running workflow obtained from 'Start Workflow' action.

Variable: variable - Output string variable for workflow status.

17.6 Wait Workflow To Complete By Name

Wait until workflow with this name started on item at this url is completed, store status of the workflow in Variable: variable1 and last status change time in Variable: variable2.

This workflow action is used to wait until site or list workflow with specified name will be completed.

Known workflow status strings are:

- Not Started
- Failed On Start
- In Progress
- Error Occurred

- Stopped By User
- Completed
- Canceled

This list may be extended by custom values, e.g. using 'Set Workflow Status' action.

Parameters:

with this name - Workflow name, e.g. 'My workflow'.
item at this url - Optional list item URL used for list workflows.
Variable: variable1 - Output string for workflow status.
Variable: variable2 - Output Date/Time variable for last status change time.

Appendix A. List of activities from Free Set

Activities from the below list are included to the free set. It means that after expiration of the Trial License, you still can use these activities to develop workflows and workflows created with its use will provide full functionality (see chapter 2 of Section II).

1. Add Array Item
2. Clear Array
3. Create Array
4. Get Array Item
5. Get Array Items Count
6. Get Index of Array Item
7. Insert Array Item
8. Remove Array Item
9. Remove Array Item at
10. Remove Duplicates from Array
11. Converts Date from Local Time to UTC
12. Converts Date from UTC to Local Time
13. Converts Date to Text
14. Evaluate Due Date
15. Get Date Property
16. Get Interval Between Dates
17. Set DateTime Constant
18. Capture All Occurrences of Text
19. Capture Text
20. Convert HTML To Text
21. Convert Number To Text
22. Convert Text Case
23. Converts Base64 to Text
24. Converts Text to Base64
25. Decode Web Text
26. Encode Web Text

27. Evaluate Mathematical Expression
28. Extract Substring
29. Format Field of Current Item
30. Get Random Number
31. Get Random String
32. Is Value Matches Regular Expression (HarePoint)
33. Join Text
34. Replace Text
35. Set Regular Expression Options
36. Split Text
37. Trim Text